
# Activity Report 2016

# Team TAMIS

## Threat Analysis and Mitigation for Information Security

Inria teams are typically groups of researchers working on the definition of a common project, and objectives, with the goal to arrive at the creation of a project-team. Such project-teams may include other partners (universities or research institutions).

RESEARCH CENTER
**Rennes - Bretagne-Atlantique**

THEME
**Security and Confidentiality**

# Table of contents

# Team TAMIS

*Creation of the Team: 2016 January 01*

**Keywords:**

**Computer Science and Digital Science:**

4. - Security and privacy
4.1. - Threat analysis
4.2. - Correcting codes
4.4. - Security of equipment and software
4.5. - Formal methods for security
4.8. - Privacy-enhancing technologies
4.9. - Security supervision

**Other Research Topics and Application Domains:**

6. - IT and telecom
6.4. - Internet of things
6.5. - Information systems
6.6. - Embedded systems
8.1. - Smart building/home
8.2. - Connected city
8.4. - Security and personal assistance
9.8. - Privacy
9.9. - Risk management
9.10. - Ethics

# 1. Members

**Research Scientists**

Axel Legay [Team leader, Inria, Researcher, HDR]
Christian Grothoff [Inria, Advanced Research position]
Annelie Heuser [CNRS, Researcher, from Oct 2016]
Olivier Zendra [Inria, Researcher, from Sep 2016]

**Faculty Members**

Fabrizio Biondi [Supelec, Associate Professor Inria "Chaire Malware" since Nov 2016; was Inria Engineer before]
Jean-Louis Lanet [Univ. of Limoges, Professor, HDR]
Kim Larsen [Univ. of Aalborg, Professor, Inria "Chaire internationale", HDR]

**Engineers**

Jeffrey Paul Burdges [Inria]
Francois-Renaud Escriva [DGA]
Ulrich Fahrenberg [Ecole Polytechnique, until Sep 2016]
Thomas Given-Wilson [Inria]
Sebastien Josse [DGA]
Colas Le Guernic [DGA]
Laurent Morin [Inria]

Van-Chan Ngo [Inria, until Feb 2016]
Jean Quilbeuf [Inria]
Sean Sedwards [Inria]
Marcello Stanisci [Inria]
Gabor Toth [Inria]
Louis-Marie Traonouez [Inria]

**PhD Students**

Sebanjila Bukasa [Inria]
Mounir Chadli [Algerian Embassy in France]
Olivier Decourbe [Inria, from Mar 2016]
Florian Dold [Inria]
Mihai Enescu [Inria, from Oct 2016]
Alvaro Garcia Recuero [Inria, until Oct 2016]
Alexandre Gonzalvez [Telecom Bretagne, from Apr 2016]
Nisrine Jafri [Inria]
Rokia Lamrani Alaoui [Inria, until Jun 2016]
Tristan Ninet [Thales, from Dec 2016, granted by CIFRE]
Aurelien Palisse [Inria]

**Post-Doctoral Fellows**

Ronan Lashermes [Inria, from Mar 2016]
Hélène Le Bouder [Inria]

**Visiting Scientists**

Razika Lounas [University of Boumerdès (Algeria), until Jun 2016]
Abdelhak Mesbah [University of Boumerdès (Algeria), Visiting PhD Student, from Apr 2016 until May 2016]
Noreddine El Janati El Idrissi [University of Rabat (Morrocco), Visiting PhD Student, from May 2016 until Jun 2016]

**Administrative Assistant**

Cecile Bouton [Inria]

**Others**

Antoine Durand [University of Bordeaux, Master 1 Student, from Jun 2016 until Sep 2016]
Gregory Durand [University of Limoges, Master Student, from Mar 2016 until Aug 2016]
Christophe Genevey-Metat [University of Rennes 1, Master Student , from May 2016 until Jul 2016]
Marouan Sami [University of Limoges, Master 1 Student, from Jul 2016 until Aug 2016]

# 2. Overall Objectives

## 2.1. Context

Security devices are subject to drastic security requirements and certification processes. They must be protected against potentially complex exploits that result from the combination of software and hardware attacks. As a result, a major effort is needed to develop new research techniques and approaches to characterize security issues, as well as to discover multi-layered security vulnerabilities in complex systems.

In recent years, we have witnessed two main lines of research to achieve this objective.

The first approach, often called *offensive security*, relies on engineering techniques and consists in attacking the system with our knowledge on its design and our past expertise. This is a creative approach that supports (1) checking whether a system is subject to existing vulnerabilities, i.e. classes of vulnerabilities that we already discovered on other systems, and (2) discovering new types of vulnerabilities that were not foreseen and that may depend on new technologies and/or programming paradigms. Unfortunately, this approach is limited to systems whose complexity remains manageable at the human level. This means that exploits which combine several vulnerabilities may be hard to identify. The second and more formal approach builds on formal models (also known as *formal methods*) to automatically detect vulnerabilities, or prove their absence. This is applicable to systems whose complexity is beyond human reasoning, but can only detect existing classes of vulnerabilities, i.e., those that have been previously characterized by offensive security.

## 2.2. Approach and motivation

The claim made by TAMIS is that *assessing security requires combining both engineering and formal techniques*.

As an example, security exploits may require combining classes of well-known vulnerabilities. The detection of such vulnerabilities can be made via formal approaches, but their successful combination requires human creativity. TAMIS's central goal is thus to demonstrably narrow the gap between the vulnerabilities found using formal verification and the issues found using systems engineering. As a second example, we point out that there are classes of attacks that exploit both the software and hardware parts of a system. Although vulnerabilities can be detected via formal methods in the software part, the impact of attacking the hardware still needs to be modeled. This is often done by observing the effect of parameter changes on the system, and capturing a model of them. To address this situation, the TAMIS team bundled resources from scalable formal verification and secure software engineering for *vulnerability analysis*, which we extend to provide methods and tools to (a) *analyze (binary) code including obfuscated malware*, and (b) *build secure systems*.

Very concrete examples better illustrate the differences and complementarity of engineering and formal techniques. First, it is well-known that formal methods can be used to detect buffer overflows. However, the definition of buffer overflows itself was made first in 1972 when the Computer Security Technology Planning study laid out the technique and claimed that over sizing could be exploited to corrupt a system. This exploit was then popularized in 1988 as one of the exploits used by the Morris worm, and only at that point systematic techniques were developed to detect it. Another example is the work we conducted in attacking smart cards. The very firsts experiments were done at the engineering level, and consisted of retrieving the key of the card in a brute force manner. Based on this knowledge, we generated user test-cases that characterize what should not happen. Later, those were used in a fully automatized model-based testing approach [71].

# 3. Research Program

## 3.1. Axis 1: Vulnerability analysis

This axis proposes different techniques to discover vulnerabilities in systems. The outcomes of this axis are (a) new techniques to discover system vulnerabilities as well as to analyze them, and (b) to understand the importance of the hardware support.

Most existing approaches used at the engineering level rely on testing and fuzzing. Such techniques consist in simulating the system for various input values, and then checking that the result conforms to a given standard. The problem being the large set of inputs to be potentially tested. Existing solutions propose to extract significant sets by mutating a finite set of inputs. Other solutions, especially concolic testing developed at Microsoft, propose to exploit symbolic executions to extract constraints on new values. We build on those existing work, and extend them with recent techniques based on dissimilarity distances and learning. We also account for the execution environment, and study techniques based on the combination of timing attacks with fuzzing techniques to discover and classify classes of behavior of the system under test.

Techniques such as model checking and static analysis have been used for verifying several types of requirements such as safety and reliability. Recently, several works have attempted to adapt model checking to the detection of security issues. It has clearly been identified that this required to work at the level of binary code. Applying formal techniques to such code requires the development of disassembly techniques to obtain a semantically well-defined model. One of the biggest issues faced with formal analysis is the state space explosion problem. This problem is amplified in our context as representations of data (such as stack content) definitively blow up the state space. We propose to use statistical model checking (SMC) of rare events to efficiently identify problematic behaviors.

We also seek to understand vulnerabilities at the architecture and hardware levels. Particularly, we evaluate vulnerabilities of the interfaces and how an adversary could use them to get access to core assets in the system. One particular mechanism to be investigated is the DMA and the so-called Trustzone. An ad-hoc technique to defend against adversarial DMA-access to memory is to keep key material exclusively in registers. This implies co-analyzing machine code and an accurate hardware model.

## 3.2. Axis 2: Malware analysis

Axis 1 is concerned with vulnerabilities. Such vulnerabilities can be exploited by an attacker in order to introduce malicious behaviors in a system. Another method to identify vulnerabilities is to analyze malware that exploits them. However, modern malware has a wide variety of analysis avoidance techniques. In particular, attackers obfuscate the code leading to a security exploit. For doing so, recent black hat research suggests hiding constants in program choices via polynomials. Such techniques hinder forensic analysis by making detailed analysis labor intensive and time consuming. The objective of research axis 2 is to obtain a full tool chain for malware analysis starting from (a) the observability of the malware via deobfuscation, and (b) the analysis of the resulting binary file. A complementary objective is to understand how hardware attacks can be exploited by malwares.

We first investigate obfuscation techniques. Several solutions exist to mitigate the packer problem. As an example, we try to reverse the packer and remove the environment evaluation in such a way that it performs the same actions and outputs the resulting binary for further analysis. There is a wide range of techniques to obfuscate malware, which includes flattening and virtualization. We will produce a taxonomy of both techniques and tools. We will first give a particular focus to control flow obfuscation via mixed Boolean algebra, which is highly deployed for malware obfuscation. We recently showed that a subset of them can be broken via SAT-solving and synthesis. Then, we will expand our research to other obfuscation techniques.

Once the malware code has been unpacked/deobfuscated, the resulting binary still needs to be fully understood. Advanced malware often contains multiple stages, multiple exploits and may unpack additional features based on its environment. Ensuring that one understands all interesting execution paths of a malware sample is related to enumerating all of the possible execution paths when checking a system for vulnerabilities. The main difference is that in one case we are interested in finding vulnerabilities and in the other in finding exploitative behavior that may mutate. Still, some of the techniques of Axis 1 can be helpful in analyzing malware. The main challenge for axis 2 is thus to adapt the tools and techniques to deal with binary programs as inputs, as well as the logic used to specify malware behavior, including behavior with potentially rare occurrences. Another challenge is to take mutation into account, which we plan to do by exploiting mining algorithms.

Most recent attacks against hardware are based on fault injection which dynamically modifies the semantics of the code. We demonstrated the possibility to obfuscate code using constraint solver in such a way that the code becomes intentionally hostile while hit by a laser beam. This new form of obfuscation opens a new challenge for secure devices where malicious programs can be designed and uploaded that defeat comprehensive static analysis tools or code reviews, due to their multi-semantic nature. We have shown on several products that such an attack cannot be mitigated with the current defenses embedded in Java cards. In this research, we first aim at extending the work on fault injection, then at developing new techniques to analyze such hostile code. This is done by proposing formal models of fault injection, and then reusing results from our work on obfuscation/deobfuscation.

### 3.3. Axis 3: Building a secure network stack

To evaluate the techniques developed in Axes 1 and 2, we analyze concrete systems developed not only with industry partners, but also within the team. By using our own systems, we can co-evolve best-practices, while externally developed systems provide realistic challenges especially with respect to analyzing obfuscated malware in the hardware or complex vulnerabilities. In this context, Christian Grothoff (ARP Inria) is currently developing a new Internet, which is supposed to be more secure. This introduces interesting challenges both in terms of vulnerability and malware analysis, and hence should be a great opportunity to mix the competences of all the members of the team.

More precisely, this system intends to challenge the idea that network security is an administrative task, where network administrators shield users with passwords, firewalls, intrusion detection systems and policies. Instead, we want to eliminate administrators that have power over user's data, and as such administrators themselves are liabilities, and because a network design that permits administrative intrusion inherently adds vulnerabilities. Instead, the system should ensure secure communication mechanisms without trusted third parties.

Key challenges we work on include (a) improving scalable secure ad-hoc decentralized routing, including key-value lookup, unicast and multicast communication, (b) protecting meta-data in the overlay using advanced decentralized onion routing, (c) a unified public-key infrastructure and identity management solution that is suitable to replace the Web-of-Trust, X.509, DNSSEC and other legacy methods for naming and identifying services, (d) secure synchronous and asynchronous messaging at scale, providing decentralized alternatives to common online social applications and addressing challenges in protocol evolution and compatibility. Finally, we are currently working on GNU Taler, a new secure privacy-preserving payment system where users never have to authenticate. This system in particular can be used as a concrete test case for the methods developed in the team.

To support this research work, we develop a framework named GNUnet. It provides a clear separation into layers, which facilitates testing and verifying the various components. However, we see that often existing formal verification techniques still do not scale to typical subsystems encountered in practice. Our objective is thus to exploit efficient and scalable formal techniques techniques proposed in Axis 1 together with engineering skills in order to guide the validation (message synchronization, data protection, ...) and reach the best compromise. An additional complication is that we need a validation process that not merely covers the software itself, but also all of its dependencies (such as database, cryptographic libraries and networking libraries). For the Taler-specific hardware, we are envisioning an NFC-powered device, which creates new challenges in terms of securing cryptographic computations in a setting where the adversary has control over the power supply. In such a case, the attacker can drive the environment and modify the behavior of the system as we have shown in Axis 2. Providing the control of the environment is a new vector for attackers.

## 4. Application Domains

### 4.1. System analysis

The work performed in Axes 1 and 2 and the methods developed there are applicable to the domain of system analysis, both wrt. program analysis and hardware analysis.

### 4.2. Cybersecurity

The work done in the 3 axes above aims at improving cybersecurity, be it via vulnerability analyses, malware analyses and the development of safer networking mechanisms.

### 4.3. Safe Internet

The work done in Axis 3 above very directly contributes to the goal of a safer Internet.

# 5. Highlights of the Year

## 5.1. Highlights of the Year

### New major release of Plasma Lab

**Participants:** Axel Legay, Sean Sedwards, Louis-Marie Traonouez.

We have released version 1.4.0 of our Plasma Lab software. This new version introduces a new command line interface for launching Plasma Lab. Besides the Graphical Interface, most of Plasma Lab functionalities are now available directly from the command line. Additionally the new version includes a new algorithm for cross entropy minimization using importance sampling. It allows to estimate the probabilities of rare events.

### Fault injection proof-of-concept

**Participants:** Axel Legay, Jean-Louis Lanet, Thomas Given-Wilson, Nisrine Jafri.

Creation of a proof of concept to show that formal verification can be used to discover fault injections induced by hardware attacks.

### Creation of LHS platform

**Participants:** Jean-Louis Lanet, Hélène Le Bouder, Ronan Lashermes.

Entry into service of the LHS platform that can be used to monitor systems, inject faults, or reason on ransomware.

### Taler Systems startup creation

**Participants:** Jeffrey Burdges, Florian Dold, Christian Grothoff, Marcello Stanisci.

A startup, Taler Systems S.A. was formally created, and we started the contractual paperwork required. An interview was given to RWGV-Genossenschaftsblatt (an internal publication of a large group of German banks).

### Contract with CISCO

**Participants:** Axel Legay, Fabrizio Biondi, Thomas Given-Wilson.

Signature of a major research collaboration contract between Tamis and CISCO to work on malware analysis. The collaboration will fund 3 engineers, trips to visit CISCO and participate to conferences on the topic, as well as a powerful servers to store and analyse malware.

### Awards

Axel Legay received the first Parnass award.

Christian Grothoff became an Ashoka fellow.

# 6. New Software and Platforms

## 6.1. MHD

GNU libmicrohttpd
KEYWORDS: Embedded - Web 2.0
SCIENTIFIC DESCRIPTION

We are providing a standards compliant and complete implementation of the HTTP server protocol that allows developers to easily write correct HTTP servers. Key challenges include code size minimization (for IoT devices), performance (zero copy, scalability to 100k concurrent connections), portability and security. MHD is already widely used in production by both academic and industrial users. Ongoing research challenges include formal verification.

FUNCTIONAL DESCRIPTION

GNU libmicrohttpd is a small C library that is supposed to make it easy to run an HTTP server as part of another application.

- Participants: Evgeny Grin, Christian Grothoff and Sree Hrsha Totakura
- Partner: The GNU Project
- Contact: Christian Grothoff
- URL: http://www.gnu.org/software/libmicrohttpd/

## 6.2. PLASMA Lab

KEYWORDS: Model Checking - Statistical - Model Checker - Runtime Analysis - Security - Code analysis - Statistics - Energy - Aeronautics

SCIENTIFIC DESCRIPTION

Statistical model checking (SMC) is a fast emerging technology for industrial scale verification and optimisation problems. Plasma was conceived to have high performance and be extensible, using a proprietary virtual machine. Since SMC requires only an executable semantics and is not constrained by decidability, we can easily implement different modelling languages and logics.

FUNCTIONAL DESCRIPTION

PLASMA Lab is a compact, efficient and flexible platform for statistical model checking of stochastic models. PLASMA Lab demonstrates the following advances: -Use your own simulator and checker via our plugin system. -Build your software around Plasma Lab using our API. -Prism (Reactives Modules Language-RML) and Biological languages supported. -Matlab and SytemC plugins. -Distributed architecture. Whether you plan to use several computers on a local area network or a grid, you can run PLASMA Lab in an easy way. -Fast algorithms. -Efficient data structure, low memory consumption. -Developed with Java for compatibility.

- Participants: Axel Legay, Sean Sedwards, Benoit Boyer, Louis-Marie Traonouez, Kevin Corre and Matthieu Simonin
- Contact: Axel Legay
- URL: http://plasma-lab.gforge.inria.fr/plasma_lab_doc/1.4.0/html/introduction.html#

## 6.3. Quail

FUNCTIONAL DESCRIPTION

Privacy is a central concern for Systems of Systems and interconnected objects. We propose QUAIL, a tool that can be used to quantify privacy of components. QUAIL is the only tool able to perform an arbitrary-precision quantitative analysis of the security of a system depending on private information. Thanks to its Markovian semantics model, QUAIL computes the correlation between the system's observable output and the private information, obtaining the amount of bits of the secret that the attacker will infer by observing the output.

- Participants: Fabrizio Biondi, Axel Legay, Louis-Marie Traonouez and Andrzej Wasowski
- Contact: Axel Legay
- URL: https://project.inria.fr/quail/

## 6.4. GNUnet

SCIENTIFIC DESCRIPTION

The GNUnet project seeks to answer the question what a modern Internet architecture should look like for a society that care about security and privacy. We are considering all layers of the existing well-known Internet, but are also providing new and higher-level abstractions (such as voting protocols, Byzantine consensus, etc.) that are today solved in application-specific ways. Research questions include the desired functionality of the overall stack, protocol design for the various layers as well as implementation considerations, i.e. how to implement the design securely.

FUNCTIONAL DESCRIPTION

GNUnet is a framework for secure peer-to-peer networking that does not use any centralized or otherwise trusted services. Our high-level goal is to provide a strong free software foundation for a global network that provides security and in particular respects privacy.

GNUnet started with an idea for anonymous censorship-resistant file-sharing, but has grown to incorporate other applications as well as many generic building blocks for secure networking applications. In particular, GNUnet now includes the GNU Name System, a privacy-preserving, decentralized public key infrastructure.

- Participants: Christian Grothoff, Florian Dold, Jeffrey Paul Burdges, Gabor Toth, Sree Hrsha Totakura and Alvaro Garcia Recuero
- Partner: The GNU Project
- Contact: Christian Grothoff
- URL: https://gnunet.org/

## 6.5. Taler

GNU Taler
KEYWORD: Privacy
SCIENTIFIC DESCRIPTION

Taler is a Chaum-style digital payment system that enables anonymous payments while ensuring that entities that receive payments are auditable. In Taler, customers can never defraud anyone, merchants can only fail to deliver the merchandise to the customer, and payment service providers can be fully audited. All parties receive cryptographic evidence for all transactions, still, each party only receives the minimum information required to execute transactions. Enforcement of honest behavior is timely, and is at least as strict as with legacy credit card payment systems that do not provide for privacy.

The key technical contribution underpinning Taler is a new refresh protocol which allows fractional payments and refunds while maintaining untraceability of the customer and unlinkability of transactions. The refresh protocol combines an efficient cut-and-choose mechanism with a link step to ensure that refreshing is not abused for transactional payments.

We argue that Taler provides a secure digital currency for modern liberal societies as it is a flexible, *libre* and efficient protocol and adequately balances the state's need for monetary control with the citizen's needs for private economic activity.
FUNCTIONAL DESCRIPTION

Taler is a new electronic payment system. It includes an electronic wallet for customers, a payment backend for merchants and the main payment service provider logic called the exchange. Taler offers Chaum-style anonymous payments for citizens, and income-transparency for taxability.

- Participants: Jeffrey Paul Burdges, Marcello Stanisci, Florian Dold, Gabor Toth and Christian Grothoff
- Partner: The GNU Project
- Contact: Christian Grothoff
- URL: http://taler.net/

## 6.6. VITRAIL - Visualisation Tool

Real-Time, Advanced, Immersive Visualization of Software / Visualizer
KEYWORD: Visualization of software
SCIENTIFIC DESCRIPTION

It is difficult for developers to explore and understand the source code of large programs, for example in objet-oriented languages programs featuring thousands of classes. Visualization methods based on daily life metaphors have thus been proposed. The VITRAIL Visualization tool (or VITRAIL Vizualizer) makes it possible to display, visualize and explore Java programs in a metaphorical way, using the city metaphor. An execution trace of the Java (byte)code provided by VITRAIL JBInstrace tool, is provided as input to VITRAIL Visualizer which displays a city-like metaphorical world showing the static structure of the code as well as some dynamic elements (calls).
This program may be used in Tamis as a basis for tools for the visualization of security events in programs.

FUNCTIONAL DESCRIPTION

This program makes it possible to displays, visualizes and explores Java programs in a metaphorical way (using the city metaphore). Useful for complex application developers/architects.

- Participants: Damien Bodenes, Olivier Zendra and Olivier Demengeon
- Contact: Olivier Zendra

## 6.7. VITRAIL 6 JBInsTrace

Real-Time, Advanced, Immersive Visualization of Software / Java Bytecode Instrumenter and Tracer
KEYWORDS: Java - Bytecode - Instrumentation - Profiling - Execution trace - Basic block
SCIENTIFIC DESCRIPTION

VITRAIL JBInsTrace is a program to instrument Java bytecode to trace its execution. The trace contains both static and dynamic information (calls). It is produced by intercepting the JVM class loader and replacing it by ours. Thus Java bytecode file are not modified, since instrumentation is performed on the fly, in memory. This makes it possible to instrument the whole program code, including libraries. Java source code is not needed. The trace which is then fed into our program VITRAIL Visualizer is an XML-like file.
This program may be used in Tamis as a basis for tools to instrument Java bytecode for security.

FUNCTIONAL DESCRIPTION

VITRAIL JBInsTrace is a program to instrument Java bytecode files to trace their execution. The trace is then fed into our VITRAIL Visualizer tool.

- Participants: Pierre Caserta and Olivier Zendra
- Contact: Olivier Zendra

## 6.8. Platforms

### *6.8.1. Malware'o'Matic*

This LHS platform is dedicated to the collect, the categorization and the analyze of malware. We are currently interested in a specific kind of malware the ransomware. The platform grabs periodically samples of public data bases, executes the ransomware without virtualization on a victim PC and evaluate the implemented detection mechanisms. Once a ransomware has been executed the image of the OS is automatically restored and a new sample is evaluated. The platform is fully automatic and target Windows platforms (seven, W10) in both 32 bits and 64 bits versions.

### *6.8.2. Faustine*

This LHS platform is dedicated to the EM fault injection experiments. It is composed of a motion table (XY), a pulse generator, an amplifier and a control PC. It injects EM pulses in a controlled way on a targeted device using an EM probe. It controls with a high precision the timing and the edges of the pulse. A recent development consists in adding a FPGA board to control the trigger in a more convenient and precise way. Then, the pulse can be triggered while a specific information is sent to the board under attack.

#### 6.8.3. EMA

This last LHS platform is dedicated to side channel analysis (SCA) for evaluating the capabilities of dynamic countermeasure developed in the ANR Cogito. This platform uses a low cost oscilloscope, an EM probe and a set of software developed to perform the analysis. An additional oscilloscope, more performant has been added to the platform to target faster devices. We use the Julia language platform and custom developments to control the target and to analyze the EM traces.

# 7. New Results

## 7.1. Results for Axis 1: Vulnerability analysis

Statistical model checking employs Monte Carlo methods to avoid the state explosion problem of probabilistic (numerical) model checking. To estimate probabilities or rewards, SMC typically uses a number of statistically independent stochastic simulation traces of a discrete event model. Being independent, the traces may be generated on different machines, so SMC can efficiently exploit parallel computation. Reachable states are generated on the fly and SMC tends to scale polynomially with respect to system description. Properties may be specified in bounded versions of the same temporal logics used in probabilistic model checking. Since SMC is applied to finite traces, it is also possible to use logics and functions that would be intractable or undecidable for numerical techniques.

Several model checking tools have added SMC as a complement to exhaustive model checking. This includes the model checker UPPAAL, for timed automata, the probabilistic model checker PRISM, and the model checker Ymer, for continuous time Markov chains. Plasma Lab [29] is the first platform entirely dedicated to SMC. Contrary to other tools, that target a specific domain and offer several analysis techniques, including basic SMC algorithms, Plasma Lab is designed as a generic platform that facilitates multiple SMC algorithms, multiple modelling and query languages and has multiple modes of use. This allows us to apply statistical model checking techniques to a wide variety of problems, reusing existing simulators. With this process we avoid the task of rewriting a model of a system in a language not ideally design to do it. This complex task often leads either to an approximation of the original system or to a more complex model harder to analyze. The task needed to support a new simulator is to implement an interface plugin between our platform Plasma Lab and the existing tool, using the public API of our platform. This task has to be performed only once to analyze all the systems supported by the existing simulator.

Plasma Lab can already be used with the PRISM language for continuous and discrete time Markov chains and biological models. During the last years we have developed several new plugins to support SystemC language [50], Simulink models [70], dynamic software architectures language [41], [14], and train interlocking systems [64]. They have been presented in recent publications.

[50] Transaction-level modeling with SystemC has been very successful in describing the behavior of embedded systems by providing high-level executable models, in which many of them have an inherent probabilistic behavior, i.e., random data, unreliable components. It is crucial to evaluate the quantitative and qualitative analysis of the probability of the system properties. Such analysis can be conducted by constructing a formal model of the system and using probabilistic model checking. However, this method is unfeasible for large and complex systems due to the state space explosion. In this paper, we demonstrate the successful use of statistical model checking to carry out such analysis directly from large SystemC models and allows designers to express a wide range of useful properties.

[70] We present an extension of the statistical model checker Plasma Lab capable of analyzing Simulink models.

[41], Dynamic software architectures emerge when addressing important features of contemporary systems, which often operate in dynamic environments subjected to change. Such systems are designed to be reconfigured over time while maintaining important properties, e.g., availability,

correctness, etc. Verifying that reconfiguration operations make the system to meet the desired properties remains a major challenge. First, the verification process itself becomes often difficult when using exhaustive formal methods (such as model checking) due to the potentially infinite state space. Second, it is necessary to express the properties to be verified using some notation able to cope with the dynamic nature of these systems. Aiming at tackling these issues, we introduce DynBLTL, a new logic tailored to express both structural and behavioral properties in dynamic software architectures. Furthermore, we propose using statistical model checking (SMC) to support an efficient analysis of these properties by evaluating the probability of meeting them through a number of simulations. In this paper, we describe the main features of DynBLTL and how it was implemented as a plug-in for PLASMA, a statistical model checker.

[14] The critical nature of many complex software-intensive systems calls for formal, rigorous architecture descriptions as means of supporting automated verification and enforcement of architectural properties and constraints. Model checking has been one of the most used techniques to automatically analyze software architectures with respect to the satisfaction of architectural properties. However, such a technique leads to an exhaustive exploration of all possible states of the system under verification, a problem that becomes more severe when verifying dynamic software systems due to their typical non-deterministic runtime behavior and unpredictable operation conditions. To tackle these issues, we propose using statistical model checking (SMC) to support the analysis of dynamic software architectures while aiming at reducing effort, computational resources, and time required for this task. In this paper, we introduce a novel notation to formally express architectural properties as well as an SMC-based toolchain for verifying dynamic software architectures described in $\pi$-ADL, a formal architecture description language. We use a flood monitoring system to show how to express relevant properties to be verified, as well as we report the results of some computational experiments performed to assess the efficiency of our approach.

[64] , accepted at HASE 2017 In the railway domain, an interlocking is the system ensuring safe train traffic inside a station by controlling its active elements such as the signals or points. Modern interlockings are configured using particular data, called application data, reflecting the track layout and defining the actions that the interlocking can take. The safety of the train traffic relies thereby on application data correctness, errors inside them can cause safety issues such as derailments or collisions. Given the high level of safety required by such a system, its verification is a critical concern. In addition to the safety, an interlocking must also ensure that availability properties, stating that no train would be stopped forever in a station, are satisfied. Most of the research dealing with this verification relies on model checking. However, due to the state space explosion problem, this approach does not scale for large stations. More recently, a discrete event simulation approach limiting the verification to a set of likely scenarios, was proposed. The simulation enables the verification of larger stations, but with no proof that all the interesting scenarios are covered by the simulation. In this paper, we apply an intermediate statistical model checking approach, offering both the advantages of model checking and simulation. Even if exhaustiveness is not obtained, statistical model checking evaluates with a parameterizable confidence the reliability and the availability of the entire system.

### *7.1.1. Verification of Dynamic Software Architectures*

**Participants:** Axel Legay, Jean Quilbeuf, Louis-Marie Traonouez.

Dynamic software architectures emerge when addressing important features of contemporary systems, which often operate in dynamic environments subjected to change. Such systems are designed to be reconfigured over time while maintaining important properties, e.g., availability, correctness, etc. $\pi$-ADL is a formal, well-founded theoretically language intended to describe software architectures under both structural and behavioral viewpoints. In order to cope with dynamicity concerns, $\pi$-ADL is endowed with architectural level primitives for specifying programmed reconfiguration operations, i.e., foreseen, pre-planned changes described at design time and triggered at runtime by the system itself under a given condition or event. Additionally, code source in the Go programming language is automatically generated from $\pi$-ADL architecture descriptions, thereby allowing for their execution.

We have developed with Plasma Lab a toolchain [14] for verifying dynamic software architectures described in $\pi$-ADL. The architecture description in $\pi$-ADL is translated towards generating source code in Go. As $\pi$-ADL architectural models do not have a stochastic execution, they are linked to a stochastic scheduler parameterized by a probability distribution for drawing the next action. Furthermore, we use existing probability distribution Go libraries to model inputs of system models as user functions. The program resulting from the compilation of the generated Go source code emits messages referring to transitions from addition, attachment, detachment, and value exchanges of architectural elements. Additionally we have introduced DynBLTL [41] a new logic tailored to express both structural and behavioral properties in dynamic software architectures.

We have developed two plugins atop the PLASMA platform, namely (i) a simulator plug-in that interprets execution traces produced by the generated Go program and (ii) a checker plugin that implements DynBLTL. With this toolchain, a software architect is able to evaluate the probability of a $\pi$-ADL architectural model to satisfy a given property specified in DynBLTL.

### *7.1.2. Statistical Model-Checking of Scheduling Systems*

**Participants:** Axel Legay, Louis-Marie Traonouez.

Cyber-Physical Systems (CPS) are software implemented control systems that control physical objects in the real world. These systems are being increasingly used in many critical systems, such as avionics and automotive systems. They are now integrated into high performance platforms, with shared resources. This motivates the development of efficient design and verification methodologies to assess the correctness of CPS.

Schedulability analysis is a major problem in the design of CPS. Software computations that implements the commands sent to the CPS are split into a set of hard real-time tasks, often periodic. These tasks are associated to strict deadlines that must be satisfied. A scheduler is responsible for dispatching a shared resource (usually CPU computation time) among the different tasks according to a chosen scheduling policy. The schedulability analysis consists in verifying that the tasks always meet their deadlines.

Over the years, the schedulability of CPS have mainly been performed by analytical methods. Those techniques are known to be effective but limited to a few classes of scheduling policies. In a series of recent work, it has been shown that schedulability analysis of CPS could be performed with a model-based approach and extensions of verification tools such as UPPAAL. It shows that such models are flexible enough to embed various types of scheduling policies that go beyond those in the scope of analytical tools.

We have extended these works to include more complex features in the design of these systems and we have experimented the use of statistical model checking as a lightweight verification technique for these systems.

We also extended the approach to statistical model checking of products lines. Our first contribution has been to propose models to design software product lines (SPL) of preemptive real-time systems [25]. Software Product Line Engineering (SPLE) allows reusing software assets by managing the commonality and variability of products. Recently, SPLE has gained a lot of attention as an approach for developing a wide range of software products from non-critical to critical software products, and from application software to platform software products.

Real-time software products (such as real-time operating systems) are a class of systems for which SPLE techniques have not drawn much attention from researchers, despite the need to efficiently reuse and customize real-time artifacts. We have proposed a formal SPLE framework for real-time systems. It focuses on the formal analysis of real-time properties of an SPL in terms of resource sharing with time dependent functionalities. Our framework provides a structural description of the variability and the properties of a real time system, and behavioral models to verify the properties using formal techniques implemented in the tools UPPAAL symbolic model checker and UPPAAL statistical model checker. For the specification of an SPL, we propose an extension of a feature model, called Property Feature Model (PFM). A PFM explicitly distinguishes features and properties associated with features, so that properties are analyzed in the context of the relevant features. We also define a non-deterministic decision process that automatically configures the products of an SPL that satisfy the constraints of a given PFM and the product conditions of customers. Finally we analyze the products against the associated properties. For analyzing real-time properties, we provide feature behavioral models

of the components of a scheduling unit, i.e. tasks, resources and schedulers. Using these feature behavioral models, a family of scheduling units of an SPL is formally analyzed against the designated properties with model checking techniques.

[25] This paper presents a formal analysis framework to analyze a family of platform products w.r.t. real-time properties. First, we propose an extension of the widely-used feature model, called Property Feature Model (PFM), that distinguishes features and properties explicitly. Second, we present formal behavioral models of components of a real-time scheduling unit such that all real-time scheduling units implied by a PFM are automatically composed to be analyzed against the properties given by the PFM. We apply our approach to the verification of the schedulability of a family of scheduling units using the symbolic and statistical model checkers of UPPAAL.

### *7.1.3. Model-based Framework for Hierarchical Scheduling Systems*

**Participants:** Axel Legay, Louis-Marie Traonouez, Mounir Chadli.

In order to reduce costs in the design of modern CPS, manufacturers devote strong efforts to maximize the number of components that can be integrated on a given platform. This can be achieved by minimizing the resource requirements of individual components. A hierarchical scheduling systems (HSS) integrates a number of components into a single system running on one execution platform. Hierarchical scheduling systems have been gaining more attention by automotive and aircraft manufacturers because they are practical in minimizing the cost and energy of operating applications.

Several papers have proposed model-based compositional framework for HSS. In [4] we proposed a methodology for optimizing the resource requirement of a component of an HSS using model checking techniques. Our methodology consists of using a lightweight statistical model checking method and a costly but absolute certain symbolic model checking method that operates on identical models.

In another work [15] we have proposed stochastic extension of HSS that allows us to capture tasks whose real-time attributes, such as deadline, execution time or period, are also characterized by probability distributions. This is particularly useful to describe mixed-critical systems and make assumptions on the hardware domain. These systems combine hard real-time periodic tasks, with soft real-time sporadic tasks. Classical scheduling techniques can only reason about worst case analysis of these systems, and therefore always return pessimistic results. Using tasks with stochastic period we can better quantify the occurrence of these tasks. Similarly, using stochastic deadlines we can relax timing requirements, and stochastic execution times are used to model the variation of the computation time needed by the tasks. These distributions can be sampled from executions or simulations of the system, or set as requirements from the specifications. For instance in avionics, display components have lower criticality. They can include sporadic tasks generated by users requests. Average user demand is efficiently modeled with a probability distribution.

We have also developed a graphical high-level language to represent scheduling units and complex hierarchical scheduling systems. In order to bridge the gap between the formalisms, we exploit Cinco, a generator for domain specific modeling tools to generate an interface between this language and the one of UPPAAL. Cinco allows to specify the features of a graphical interface in a compact meta-model language. This is a flexible approach that could be extended to any formal model of scheduling problem.

[4] Compositional reasoning on hierarchical scheduling systems is a well-founded formal method that can construct schedulable and optimal system configurations in a compositional way. However, a compositional framework formulates the resource requirement of a component, called an interface, by assuming that a resource is always supplied by the parent components in the most pessimistic way. For this reason, the component interface demands more resources than the amount of resources that are really sufficient to satisfy sub-components. We provide two new supply bound functions which provides tighter bounds on the resource requirements of individual components. The tighter bounds are calculated by using more information about the scheduling system. We evaluate our new tighter bounds by using a model-based schedulability framework for hierarchical scheduling systems realized as UPPAAL models. The timed models are checked using model checking tools UPPAAL and UPPAAL SMC, and we compare our results with the state of the art tool CARTS.

[15] Over the years, schedulability of Cyber-Physical Systems (CPS) have mainly been performed by analytical methods. Those techniques are known to be effective but limited to a few classes of scheduling policies. In a series of recent work, we have shown that schedulability analysis of CPS could be performed with a model-based approach and extensions of verification tools such as UPPAAL. One of our main contribution has been to show that such models are flexible enough to embed various types of scheduling policies that go beyond those in the scope of analytical tools. In this paper, we go one step further and show how our formalism can be extended to account for stochastic information, such as sporadic tasks whose attributes depend on the hardware domain. Our second contribution is to make our tools accessible to average users that are not experts in formal methods. For doing so, we propose a graphical and user-friendly language that allows us to describe scheduling problems. This language is automatically translated to formal models by exploiting a meta-model approach. The principle is illustrated on a case study.

### *7.1.4. Verification of Interlocking Systems*

**Participants:** Axel Legay, Louis-Marie Traonouez, Jean Quilbeuf.

An interlocking is a system that controls the train traffic by acting as an interface between the trains and the railway track components. The track components are for example, the signals that allow the train to proceed, or the points that guide the trains from one track to another. The paths followed by the trains are called routes. Modern interlockings are computerized systems that are composed of generic software and application data.

We have proposed in collaboration with Université Catholique de Louvain and Alstom a method to automatically verify an interlocking using simulation and statistical model checking [64]. We use a simulator developed by Université Catholique de Louvain that is able to generate traces of the interlocking systems from a track layout and application data. This simulator is plug with Plasma Lab using a small interface developed with Plasma Lab's API. Then, the traces generated by the simulator have been used by Plasma Lab SMC algorithms to measure the correctness of the system. We have used Monte-Carlo and importance splitting algorithms to verify this system.

### *7.1.5. Advanced Statistical Model Checking*

**Participants:** Axel Legay, Sean Sedwards, Louis-Marie Traonouez.

Statistical model checking (SMC) addresses the state explosion problem of numerical model checking by estimating quantitative properties using simulation. Rare events, such as software bugs, are often critical to the performance of systems but are infrequently observed in simulations. They are therefore difficult to quantify using SMC. Nondeterministic systems deliberately leave parts of system behaviour undefined, hence it is not immediately possible to simulate them. Our ongoing work thus pushes the boundaries of the cutting edge of SMC technology by focusing on rare event verification and the optimisation of nondeterminism.

#### *7.1.5.1. Optimizing Nondeterministic Systems*

[17] Probabilistic timed automata (PTA) generalize Markov decision processes (MDPs) and timed automata (TA), both of which include nondeterminism. MDPs have discrete nondeterministic choices, while TA have continuous nondeterministic time. In this work we consider finding *schedulers* that resolve all nondeterministic choices in order to maximize or minimize the probability of a time-bounded LTL property. Exhaustive numerical approaches often fail due to state explosion, hence we present a new lightweight on-the-fly algorithm to find near-optimal schedulers. To discretize the continuous choices we make use of the classical region and zone abstractions from timed automata model checking. We then apply our recently developed "smart sampling" technique for statistical verification of Markov decision processes. On standard case studies our algorithm provides good estimates for both maximum and minimum probabilities. We compare our new approach with alternative techniques, first using tractable examples from the literature, then motivate its scalability using case studies that are intractable to numerical model checking and challenging for existing statistical techniques.

#### *7.1.5.2. Rare Event Verification*

[3] Importance sampling is a standard technique to significantly reduce the computational cost of quantifying rare properties of probabilistic systems. It works by weighting the original distribution of the system to make the rare property appear more frequently in simulations, then compensating the resulting estimate by the weights. This can be done on the fly with minimal storage, but the challenge is to find *near optimal* importance sampling distributions efficiently, where optimal means that paths that do not satisfy the property are never seen, while paths that satisfy the property appear in the same proportion as in the original distribution.

Our approach uses a tractable cross-entropy minimization algorithm to find an optimal parameterized importance sampling distribution. In contrast to previous work, our algorithm uses a naturally defined low dimensional vector to specify the distribution, thus avoiding an explicit representation of a transition matrix. Our parametrisation leads to a unique optimum and is shown to produce many orders of magnitude improvement in efficiency on various models. In this work we specifically link the existence of optimal importance sampling distributions to time-bounded logical properties and show how our parametrisation affects this link. We also motivate and present simple algorithms to create the initial distribution necessary for cross-entropy minimization. Finally, we discuss the open challenge of defining error bounds with importance sampling and describe how our optimal parameterized distributions may be used to infer qualitative confidence.

[10] In this work we consider rare events in systems of Stochastic Timed Automata (STA) with time-bounded reachability properties. This model may include rarity arising from explicit discrete transitions, as well as more challenging rarity that results from the intersection of timing constraints and continuous distributions of time. Rare events have been considered with simple combinations of continuous distributions before, e.g., in the context of queueing networks, but we present an automated framework able to work with arbitrarily composed STA. By means of symbolic exploration we first construct a zone graph that excludes unfeasible times. We then simulate the system within the zone graph, avoiding "dead ends" on the fly and proportionally redistributing their probability to feasible transitions. In contrast to many other importance sampling approaches, our "proportional dead end avoidance" technique is guaranteed by construction to reduce the variance of the estimator with respect to simulating the original system. Our results demonstrate that in practice we can achieve substantial overall computational gains, despite the symbolic analysis.

[49] In this invited paper we outline some of our achievements in quantifying rare properties in the context of SMC. In addition to the importance sampling techniques described above, we also describe our work on importance *splitting*. Importance splitting works by decomposing the probability of a rare property into a product of probabilities of sub-properties that are easier to estimate. The sub-properties are defined by *levels* of a *score function* that maps states of the system$\times$property product automaton to values. We have provided the first general purpose implementation of this approach, using user-accessible "observers" that are compiled automatically from the property. These observers may be used by both fixed and adaptive level importance splitting algorithms and are specifically designed to make distribution efficient.

### *7.1.6. Side-channel Analysis of Cryptographic Substitution Boxes*

**Participants:** Axel Legay, Annelie Heuser.

With the advent of the Internet of Things, we are surrounded with smart objects (aka things) that have the ability to communicate with each other and with centralized resources. The two most common and widely noticed artefacts are RFID and Wireless Sensor Networks which are used in supply-chain management, logistics, home automation, surveillance, traffic control, medical monitoring, and many more. Most of these applications have the need for cryptographic secure components which inspired research on cryptographic algorithms for constrained devices. Accordingly, lightweight cryptography has been an active research area over the last 10 years. A number of innovative ciphers have been proposed in order to optimize various performance criteria and have been subject to many comparisons. Lately, the resistance against side-channel attacks has been considered as an additional decision factor.

Side-channel attacks analyze physical leakage that is unintentionally emitted during cryptographic operations in a device (e.g., power consumption, electromagnetic emanation). This side-channel leakage is statistically dependent on intermediate processed values involving the secret key, which makes it possible to retrieve the secret from the measured data.

Side-channel analysis (SCA) for lightweight ciphers is of particular interest not only because of the apparent lack of research so far, but also because of the interesting properties of substitution boxes (S-boxes). Since the nonlinearity property for S-boxes usually used in lightweight ciphers (i.e., $4 \times 4$) can be maximally equal to 4, the difference between the input and the output of an S-box is much smaller than for instance for AES. Therefore, one could conclude that from that aspect, SCA for lightweight ciphers must be more difficult. However, the number of possible classes (e.g., Hamming weight (HW) or key classes) is significantly lower, which may indicate that SCA must be easier than for standard ciphers. Besides the difference in the number of classes and consequently probabilities of correct classification, there is also a huge time and space complexity advantage (for the attacker) when dealing with lightweight ciphers.

In [23], [67] we give a detailed study of lightweight ciphers in terms of side-channel resistance, in particular for software implementations. As a point of exploitation we concentrate on the non-linear operation (S-box) during the first round. Our comparison includes SPN ciphers with 4-bit S-boxes such as KLEIN, PRESENT, PRIDE, RECTANGLE, Mysterion as well as ciphers with 8-bit S-boxes: AES, Zorro, Robin. Furthermore, using simulated data for various signal-to-noise ratios (SNR) we present empirical results for Correlation Power Analysis (CPA) and discuss the difference between attacking 4-bit and 8-bit S-boxes.

Following this direction current studies evaluate and connect cryptographic properties with side-channel resistance. More precisely, in an ideal setting a cipher should be resilient against cryptanalyses as well as side-channel attacks and yet easy and cheap to be implemented. However, since that does not seem to be possible at the current level of knowledge, we are required to make a number of trade-offs. Therefore, we investigate several S-box parameters and connect them with well known cryptographic properties of S-boxes. Moreover, when possible we give clear theoretical bounds on those parameters as well as expressions connecting them with properties like nonlinearity and $\delta$-uniformity. We emphasize that we select the parameters to explore on the basis of their possible connections with the side-channel resilience of S-boxes.

To this end, we divide the primary goal into several sub-problems. First, we discuss what is the maximal number of fixed points one can have in an optimal S-box. The question of the maximal number of fixed points for an optimal S-box is of interest on its own, but also in the side-channel context since intuitively an S-box with many fixed points should consume less power and therefore have less leakage. Moreover, the preservation of Hamming weight and a small Hamming distance between $x$ and $F(x)$ are two more properties each of which could strengthen the resistance to SCA. Our findings show that notably in the case when exactly preserving the Hamming weight, the confusion coefficient reaches good value and consequently the S-box has good SCA resilience. We show that the S-boxes with no differences in the Hamming weight of their input and output pairs (and even, S-boxes $F$ such that $F(x)$ have Hamming weight near the Hamming weight of $x$, on average) or S-boxes such that $F(x)$ lies at a small Hamming distance from $x$ cannot have high nonlinearity (although the obtainable values are not too bad for $n = 4, 8$) and therefore are not attractive in practical applications. Note that an S-box with many fixed points is also a particular case of an S-box that preserves the Hamming weight/distance between the inputs and outputs. Furthermore, our study includes involutive functions since they have a particular advantage over general pseudo-permutations. In particular, not only from an implementation viewpoint but also their side-channel resilience is the same regardless if an attacker considers the encryption or decryption phase as well as attacking the first or the last round. Next, we find a theoretical expression connecting the confusion coefficient with that of preserving the Hamming weight of inputs and outputs.

In the practical part, we first confirm our theoretical findings about the connection between preserving Hamming weight and the confusion coefficient. Besides that, we give a number of S-box examples of size $4 \times 4$ intended to provide more insight into possible trade-offs between cryptographic properties and side-channel resilience. However, our study shows that mostly preserving Hamming weight might not automatically result in a small minimum confusion coefficient and thus in higher side-channel resistance. We therefore in

detail examine the influence of $F$ on the confusion coefficient in general by concentrating on the input (in which key hypothesis are made) and the minimum value of the confusion coefficient. Following, we evaluate a number of S-boxes used in today's ciphers and show that their SCA resilience can significantly differ. Finally, we point out that non-involutive S-boxes might bring a significant disadvantage in case an attacker combines the information about $F$ and $F^{-1}$ by either targeting both first and last round of an algorithm or encryption and decryption.

[67] Side-channel Analysis of Lightweight Ciphers: Current Status and Future Directions

[23] Side-channel Analysis of Lightweight Ciphers: Does Lightweight Equal Easy?

### *7.1.7. Binary Code Analysis: Formal Methods for Fault Injection Vulnerability Detection*

**Participants:** Axel Legay, Thomas Given-Wilson, Nisrine Jafri, Jean-Louis Lanet.

Formal methods such as model checking provide a powerful tool for checking the behaviour of a system. By checking the properties that define correct system behaviour, a system can be determined to be correct (or not).

Increasingly fault injection is being used as both a method to attack a system by a malicious attacker, and to evaluate the dependability of the system. By finding fault injection vulnerabilities in a system, the resistance to attacks or faults can be detected and subsequently addressed.

A process is presented that allows for the automated simulation of fault injections. This process proceeds by taking the executable binary for the system to be tested, and validating the properties that represent correct system behaviour using model checking. A fault is then injected into the executable binary to produce a mutant binary, and the mutant binary is model checked also. A different result to the validation of the executable binary in the checking of the mutant binary indicates a fault injection vulnerability.

This process has been automated with existing tools, allowing for easy checking of many different fault injection attacks and detection of fault injection vulnerabilities. This allows for the detection of fault injection vulnerabilities to be fully automated, and broad coverage of the system to be formally shown.

### *7.1.8. Security at the hardware and software boundaries*

**Participants:** Axel Legay, Nisrine Jafri, Jean-Louis Lanet, Ronan Lashermes, Hélène Le Bouder.

#### *7.1.8.1. IoT security*

IoT security has to face all the challenges of the mainstream computer security but also new threats. When an IoT device is deployed, most of the time it operates in a hostile environment, i.e. the attacker can perform any attack on the device. If secure devices use tamper resistant chip and are programmed in a secure manner, IoT use low cost micro-controllers and are not programmed in a secure way. We developed new attacks but also evaluate how the code polymorphism can be used against these attacks. In [45] [27] we developed a template attack to retrieve the value of a PIN code from a cellphone. We demonstrated that the maximum trials to retrieve the four bytes of secret PIN is 8 and in average 3 attempts are sufficient. A supervised learning algorithm is used.

Often smart phones allow up to 10 attempts before locking definitely the memory. We used an embedded code generator [16], [45] dedicated to a given security function using a DSL to increase the security level of a non tamper resistant chip. We brought to the fore that a design of the software for protecting against fault attacks decreases the security against SCA. Fault attack is a mean to execute a code that is slightly different from the one that has been loaded into the device. Thus, to be sure that a genuine code cannot be dynamically transformed, one needs to analyze any possibility of a code to be transformed.

The work presented in [34] made possible to design an extremely effective architecture to achieve Montgomery modular multiplication. The proposed solution combines a limited resource consumption with the lowest latency compared with the literature. This allows to envisage new applications of asymmetric cryptography in systems with few resources. In order to find a cryptographic key using hidden channels, most attacks use the a priori knowledge of texts sent or received by the target. The proposed analysis presented in [28] does not use these assumptions. A belief propagation technique is used to cross the information obtained from leaked information with the equations governing the targeted algorithm.

*7.1.8.2. Safe update mechanism for IoT*

One of the challenges for IoT is the possibility to update the code through a network. This is done by stopping the system, loading the new version, verifying the signature of the firmware and installing it into the memory. Then, the memory must be cleaned to eliminate the code and the data of the previous version. Some IoT (sensor acquisition and physical system control) requires to never stop while executing the code. We have developed a complete architecture that performs such an update with real time capabilities. If one wants to use this characteristic in a real world it should pass certification. In particular he has to demonstrate that the system performs as expected. We used formal methods (mainly Coq) to demonstrate that the semantics of the code is preserved during the update. In [30], we paid attention to the detection of the Safe Update Point (SUP) because our implementation had some time an unstable behavior. We demonstrated that in a specific case, while several threads using code to be updated, the system enters into a deadlock. After discovering the bug, we patched our system.

*7.1.8.3. Reverse engineering of firmware*

Reverse engineering has two aspects; code reverse for which the literature is abundant and data reverse i.e. understanding the meaning of a structure and its usage has been less studied. The first step in reverse engineering consists in getting access of the code. In the case of romized code in a SoC, the access to the code is protected by a MMU mechanism and thus is an efficient mitigation mechanism against reverse engineering. In [8], [2] and [33] we set up several attacks to get access to the code even in presence of a MMU. The attack in [8] uses a vulnerability in the API where an object can be used instead of an array. This allows to read and write the code area leading to the possibility to execute arbitrary code in memory. In [33], we use the attack tree paradigm to explore all the possibilities to mount an attack on a given product. In [2], we used a ROP (Return Oriented Programming) attack to inject a shell code in the context of another application. Due to the fact that the shell code is executed in the context of the caller, the firewall is unable to detect the access to the secure container of the targeted application. This allows us to retrieve the content of the secure containers.

Once the dump has been obtained, one can try to retrieve code and data. Retrieving code is not obvious but several tools exist to help the analyst. These tools require that all the ISA (Instruction Set Architecture) is known. Sometime, the ISA is not known and in particular when one wants to obfuscate the code, he can use a virtual machine to execute dedicated byte code. In [32], we developed a methodology to infer the missing byte code, then we execute a data type inference to understand the memory management algorithm.

## 7.2. Results for Axis 2: Malware analysis

The detection of malicious programs is a fundamental step to be able to guarantee system security. Programs that exhibit malicious behavior, or *malware*, are commonly used in all sort of cyberattacks. They can be used to gain remote access on a system, spy on its users, exfiltrate and modify data, execute denial of services attacks, etc.

Significant efforts are being undertaken by software and data companies and researchers to protect systems, locate infections, and reverse damage inflicted by malware. Malware analysis can be divided in the following three main problems:

### *7.2.1. Malware Detection*

**Participants:** Axel Legay, Fabrizio Biondi, Olivier Decourbe, Mike Enescu, Thomas Given-Wilson, Annelie Heuser, Nisrine Jafri, Jean-Louis Lanet, Jean Quilbeuf.

Given a file or data stream, the malware detection problem consists of understanding if the file or data stream contain traces of malicious behavior. For binary executable files in particular, this requires reverse engineering the file's behavior to understand if it is malicious. The main reverse engineering techniques are categorized as:

**Static Analysis** This refers to techniques that analyze the file without executing it. It includes disassembling the file's executable code and analyzing other static features of the binary, like its import/export table, hash, etc. The file's control flow and system flow graphs can be retrieved statically (unless they are obfuscated; see below) and used to guide the exploration of the file's semantics in the search of

malicious behavior. Information flow can be tracked since hostile applications often try to transmit private information to distant servers (this form of malware are now widely spread in the mobile world). The challenge consists in detecting into a file that a private information does not leak to the external world. The verification can be done statically, dealing with storage channel (implicit or explicit), but not with side channel.

**Dynamic Analysis** This refers to techniques that actually executed the file in a sandbox (usually a virtualized environment) and analyze its interaction with the sandbox. This technique is effective in understanding the file's actual interactions with the system, making it easy to detect malicious behavior. However, malware often implements sandbox detection techniques to detect when it is being run in a virtualized environment, when functions or system calls are hooked by the analyst, or when the sandbox does not look like a normal user's machine (e.g. because it does not contain any document). Dynamic tracking of information flow makes it possible to cope with side channel attacks. With temporal side channel, the challenge lies in the potential declassification procedure used by malware to escape the analysis. We extend the TaintDroid framework to cope with native code invocation [47]. This approach reduces the false positive warning drastically. Recently we have extended this work to cope with timing side channels [under submission]. We are developing a new malware that declassifies the labels thanks to the audio system of the smart-phone. This is a joint work with Telecom Bretagne.

**Hybrid Analysis** This refers to technique that combine both static and dynamic behavior, i.e. both code analysis and execution. While more complex to implement, these techniques are able to overcome many of the shortcomings of full static and full dynamic analysis. The best example of a hybrid technique is concolic (a portmanteau for CONCrete + symbOLIC) analysis.

To contribute to concolic analysis, we are working on the state-of-the-art `angr` concolic execution engine to make it fast and efficient enough to analyze large executable malware files efficiently. We are improving `angr` 's parallelism and allowing it to precompute semantic stubs of function and system calls, allowing it to focus its analysis on the main file without having to branch in the rest of the operative system. We plan to contribute our improvements to the main `angr` branch, so that the whole community can benefit from them.

### *7.2.2. Malware Deobfuscation*

**Participants:** Axel Legay, Fabrizio Biondi, Olivier Decourbe, Mike Enescu, Thomas Given-Wilson, Annelie Heuser, Nisrine Jafri, Jean-Louis Lanet, Jean Quilbeuf.

Given a file (usually a portable executable binary or a document supporting script macros), deobfuscation refers to the preparation of the file for the purposes of further analysis. Obfuscation techniques are specifically developed by malware creators to hinder detection reverse engineering of malicious behavior. Some of these techniques include:

**Packing** Packing refers to the transformation of the malware code in a compressed version to be dynamically decompressed into memory and executed from there at runtime. Packing techniques are particularly effective against static analysis, since it is very difficult to determine statically the content of the unpacked memory to be executed, particularly if packing is used multiple times. The compressed code can also be encrypted, with the key being generated in a different part of the code and used by the unpacking procedure, or even transmitted remotely from a command and control (C&C) server.

**Control Flow Flattening** This technique aims to hinder the reconstruction of the control flow of the malware. The malware's operation are divided into basic blocks, and a dispatcher function is created that calls the blocks in the correct order to execute the malicious behavior. Each block after its execution returns control to the dispatcher, so the control flow is flattened to two levels: the dispatcher above and all the basic blocks below.

To prevent reverse engineering of the dispatcher, it is often implemented with a cryptographic hash function. A more advanced variant of this techniques embed a full virtual machine with a randomly generated instruction set, a virtual program counted, and a virtual stack in the code, and uses the machine's interpreter as the dispatcher.

Virtualization is a very effective technique to prevent reverse engineering. To contrast it, we are implementing state-of-the-art devirtualization algorithms in `angr` , allowing it to detect and ignore the virtual machine code and retrieving the obfuscated program logic. Again, we plan to contribute our improvements to the main `angr` branch, thus helping the whole security community fighting virtualized malware.

**Opaque Constants and Conditionals** Reversing packing and control flow flattening techniques requires understanding of the constants and conditionals in the program, hence many techniques are deployed to obfuscate them and make them unreadable by reverse engineering techniques. Such techniques are used e.g. to obfuscate the decryption keys of packed encrypted code and the conditionals in the control flow.

We have proven the efficiency of dynamic synthesis in retrieving opaque constant and conditionals, compared to the state-of-the-art approach of using SMT (Satisfiability Modulo Theories) solvers, when the input space of the opaque function is small enough. We are developing techniques based on fragmenting and analyzing by brute force the input space of opaque conditionals, and SMT constraints in general, to be integrated in SMT solvers to improve their effectiveness.

### *7.2.3. Malware Classification*

**Participants:** Axel Legay, Fabrizio Biondi, Olivier Decourbe, Mike Enescu, Thomas Given-Wilson, Annelie Heuser, Nisrine Jafri, Jean-Louis Lanet, Jean Quilbeuf.

Once malicious behavior has been located, it is essential to be able to classify the malware in its specific family to know how to disinfect the system and reverse the damage inflicted on it.

While it is rare to find an actually previously unknown malware, morphic techniques are employed by malware creators to ensure that different generations of the same malware behave differently enough than it is hard to recognize them as belonging to the same family. In particular, techniques based on the syntax of the program fails against morphic malware, since syntax can be easily changed.

To this end, semantic signatures are used to classify malware in the appropriate family. Semantic signatures capture the malware's behavior, and are thus resistant to morphic and differentiation techniques that modify the malware's syntactic signatures. We are investigating semantic signatures based on the program's System Call Dependency Graph (SCDG), which have been proven to be effective and compact enough to be used in practice. SCDGs are often extracted using a technique based on pushdown automata that is ineffective against obfuscated code; instead, we are applying concolic analysis via the `angr` engine to improve speed and coverage of the extraction.

Once a semantic signature has been extracted, it has to be compared against large database of known signatures representing the various malware families to classify it. The most efficient way to obtain this is to use a supervised machine learning classifier. In this approach, the classifier is trained with a large sample of signatures malware annotated with the appropriate information about the malware families, so that it can learn to quickly and automatically classify signatures in the appropriate family. Our work on machine learning classification focuses on using SCDGs as signatures. Since SCDGs are graphs, we are investigating and adapting algorithms for the machine learning classification of graphs, usually based on measures of shared subgraphs between different graphs.

In malware detection and classification, it is fundamental to have a false positive rate (i.e. rate of cleanware classified as malware) approaching zero, otherwise the classification system will classify hundred or thousands of cleanware files as malware, making it useless in practice. To decrease the false positive rate, the classifier is also trained with a large and representative database of cleanware, so that it can discriminate between signatures of cleanware and malware with a minimal false positive rate. We use a large database of malware and cleanware to train our classifier, thus guaranteeing a high detection rate with a small false positive rate.

### *7.2.4. Papers*

This section gathers papers that are results common to all sections above pertaining to Axis 2.

[57] Black-box synthesis is more efficient than SMT deobfuscation on predicates obfuscated with Mixed-Boolean Arithmetics.

[66] Recently fault injection has increasingly been used both to attack software applications, and to test system robustness. Detecting fault injection vulnerabilities has been approached with a variety of methods, yielding varied results. This paper proposes a general process using model checking to detect fault injection vulnerabilities in binaries. The process is implemented and used to detect a variety of different kinds of fault injection vulnerabilities in binaries.

[59] Fault-injection exploits hardware weaknesses to perturbate the behaviour of embedded devices. Here, we present new model-based techniques and tools to detect such attacks developed at the High-Security Laboratory at Inria.

[52] We proposed to use a bare metal approach without virtualization and a method to let the system stop the execution while the malware has been deployed in memory.

[51] We present our framework to grab sample from the net, evaluate it on victim PC and detect its presence thanks to our counter measures.

[53] In this paper, two counter measures are presented. The first one is related with the mode ECB of the AES cryptographic algorithm and the second is related with the usage of the crypto API. We developed a cryptographic provider which intercepts the key generation and store it in a safe place. Then we are able to decipher any files that the malware should have encrypted.

## 7.3. Results for Axis 3: Building a secure network stack

### *7.3.1. Private set intersection cardinality*

**Participants:** Jeffrey Burdges, Alvaro Garcia Recuero, Christian Grothoff.

We designed new efficient protocol for privacy-preserving signed set intersection cardinality using blinded BLS signatures over bilinear maps and demonstrated its utility in machine learning for abuse detection in decentralised online social networks. The paper was presented at DPM 2016 [21].

### *7.3.2. Cell tower privacy*

**Participants:** Christian Grothoff, Neal Walfield.

We analyzed real-world mobility data based on cell tower traces, and illustrated how cell tower trace data can be used to identify patterns of life. We then used these results to predict future locations over a 24h period in 15 minute intervals with 80% accuracy [43].

### *7.3.3. Taler protocol improvements*

**Participants:** Jeffrey Burdges, Florian Dold, Christian Grothoff, Marcello Stanisci.

We improved the Taler payment system protocol [13] to (1) reduce storage requirements for the exchange, which was the dominant cost, and (2) reduce security assumptions by avoiding the use of AES entirely.

We adapted the payment handshake to work even if JavaScript is disabled for the Web page, and adjusted the protocol to match discussions for future Web payment protocols from W3c. The protocol was extended with accounting functions to allow merchants to trace payments for their back office requirements. The user interface of the Taler wallet was streamlined, the wallet can finally get change, and the extension was made to work with Firefox. A public demonstrator was launched at https://demo.taler.net/.

## 7.4. Other research results: Information-Theoretical Quantification of Security Properties

**Participants:** Axel Legay, Fabrizio Biondi, Mounir Chadli, Thomas Given-Wilson.

Information theory provides a powerful quantitative approach to measuring security and privacy properties of systems. By measuring the *information leakage* of a system, security properties can be quantified, validated, or falsified. When security concerns are non-binary, information theoretic measures can quantify exactly how much information is leaked. The knowledge of such information is strategic in the developments of component-based systems.

The quantitative information-theoretical approach to security models the correlation between the secret information of the system and the output that the system produces. Such output can be observed by the attacker, and the attacker tries to infer the value of the secret information by combining this information with their prior knowledge of the system.

Armed with the produced output of the system, the attacker tries to infer information about the secret information that produced the output. The quantitative analysis we consider defines and computes how much information the attacker can expect to infer (typically measured in bits). This expected leakage of bits is the information leakage of the system.

The quantitative approach generalizes the qualitative approach and thus provides superior analysis. In particular, a system respects non-interference if and only if its leakage is equal to zero. In practice very few systems respect non-interference, and for those that don't it is imperative to be able to distinguish between the systems leaking very small amounts of secret information and systems leaking a significant amount of secret information, since only the latter are considered to pose a security vulnerability to the system.

Applied to shared-key cryptosystems, this approach allows precise reasoning about the information leakage of the secret key when the attacker knows the encoder function and information about the distribution of messages. In such scenarios, this work has generalised perfect secrecy, and so provides a more useful measure for unconditional cryptosystems (results that are safe against future advances in computing capabilities and theoretical breakthroughs in unsolved problems).

This work also explored scenarios where the attacker has less information about the cryptosystem; such as not knowing the encoder function, or not knowing the message distribution. Results here formalised that the attacker can never improve their attacks by having bad prior information, thus ensuring misinformation is always useful. Also, results show that the choice of encoder function may strengthen the cryptosystem against being learned by the attacker through observation. In particular, we showed that a well designed encoder function (represented as a matrix) has an infinitude of freedom for the attacker. Thus, the attacker cannot accurately learn all the secret information merely by observation.

There are several different scenarios where the attacker is trying to learn the secret information about the system. Here this is explored by considering what the secret information is, or equivalently, what prior knowledge the attacker has about the system.

Our new results in information leakage computation include implementing a hybrid precise-statistical computation algorithm for our QUAIL tool. The new algorithm bridges the gap between statistical and formal techniques by using static program analysis to extract structural information about the program to be analyze and decide whether each part of it would be analyzed more efficiently with precise or statistical analysis. Then each part is analyzed with the most appropriate technique, and all analyses are combined into a final result. This new hybrid method outperforms precise and statistical analysis in computation time and precision, and is a clear example of the advantages of combining precise and statistical techniques.We refer to the tools section for more details.

Additionally, we have considered how the scheduling of privileged and unprivileged processes on a shared memory could allow an unprivileged process to access confidential information temporarily stored in the memory by a privileged process. This is for instance the case in cache attacks. We have developed a general model of information leakage for scheduled systems. Our model considers a finer granularity than previous attempts on the subject, allowing us to schedule processes with small leakage, and schedule sets of processes that were considered unschedulable with no leakage by the state of the art.

[1] Preserving the privacy of private communication is a fundamental concern of computing addressed by encryption. Information-theoretic reasoning models unconditional security where the strength of

the results does not depend on computational hardness or unproven results. Usually the information leaked about the message by the ciphertext is used to measure the privacy of a communication , with perfect secrecy when the leakage is 0. However this is hard to achieve in practice. An alternative measure is the equivocation, intuitively the average number of message/key pairs that could have produced a given cipher-text. We show a theoretical bound on equivocation called max-equivocation and show that this generalizes perfect secrecy when achievable, and provides an alternative measure when perfect secrecy is not achievable. We derive bounds for max-equivocation for symmetric encoder functions and show that max-equivocation is achievable when the entropy of the ciphertext is minimized. We show that max-equivocation easily accounts for key re-use scenarios, and that large keys relative to the message perform very poorly under equivocation. We study encoders under this new perspective, deriving results on their achievable maximal equivocation and showing that some popular approaches such as Latin squares are not optimal. We show how unicity attacks can be naturally modeled, and how relaxing encoder symmetry improves equivocation. We present some algorithms for generating encryption functions that are practical and achieve 90 to 95% of the theoretical best, improving with larger message spaces.

[24] Analysis of a probabilistic system often requires to learn the joint probability distribution of its random variables. The computation of the exact distribution is usually an exhaustive precise analysis on all executions of the system. To avoid the high computational cost of such an exhaustive search, statistical analysis has been studied to efficiently obtain approximate estimates by analyzing only a small but representative subset of the system's behavior. In this paper we propose a hybrid statistical estimation method that combines precise and statistical analyses to estimate mutual information and its confidence interval. We show how to combine the analyses on different components of the system with different precision to obtain an estimate for the whole system. The new method performs weighted statistical analysis with different sample sizes over different components and dynamically finds their optimal sample sizes. Moreover it can reduce sample sizes by using prior knowledge about systems and a new abstraction-then-sampling technique based on qualitative analysis. We show the new method outperforms the state of the art in quantifying information leakage.

[12] The protection of users' data conforming to best practice and legislation is one of the main challenges in computer science. Very often, large-scale data leaks remind us that the state of the art in data privacy and anonymity is severely lacking. The complexity of modern systems make it impossible for software architect to create secure software that correctly implements privacy policies without the help of automated tools. The academic community needs to invest more effort in the formal modeling of security and anonymity properties, providing a deeper understanding of the underlying concepts and challenges and allowing the creation of automated tools to help software architects and developers. This research track provides numerous contributions to the formal modeling of security and anonymity properties and the creation of tools to verify them on large-scale software projects.

[62] High-security processes typically have to load confidential information, such as encryption keys or private data, into memory as part of their operation. In systems with a single shared memory, when high-security processes are switched out due to context switching, confidential information may remain in memory and be accessible to low-security processes. This paper considers this problem from the perspective of scheduling. A formal model supporting preemption is introduced that allows: reasoning about leakage between high-and low-security processes, and producing information-leakage aware schedulers. Several information-leakage aware heuristics are presented in the form of compositional pre-and postprocessors as part of a more general scheduling approach. The effectiveness of such heuristics is evaluated experimentally, showing them to achieve significantly better schedulability than the state of the art.

# 8. Bilateral Contracts and Grants with Industry

## 8.1. Bilateral Contracts with Industry

- CISCO (http://www.cisco.com)
- Thales (https://www.thalesgroup.com)
- Oberthur Technologies (http://www.oberthur.com/)

# 9. Partnerships and Cooperations

## 9.1. Regional Initiatives

AREDgrant for Nisrine Jafri.

Postdocs grants for Fabrizio Biondi, Jeffrey Paul Burdges, Florian Dold, Ronan Lashermes.

## 9.2. National Initiatives

### *9.2.1. ANR*

- ANR MALTHY, Méthodes ALgèbriques pour la vérification de modèles Temporisés et HYbrides, Thao Dang, 4 years, Inria and VISEO and CEA and VERIMAG
- ANR COGITO, Runtime Code Generation to Secure Devices,, 3 years, Inria and CEA and ENSMSE and XLIM.

## 9.3. European Initiatives

### *9.3.1. FP7 & H2020 Projects*

#### *9.3.1.1. ACANTO*

**Participants:** Axel Legay, Thomas Given-Wilson, Sean Sedwards, Olivier Zendra.

Start: 2015. End: 2018.

The population of the advanced countries is ageing. This simple and widely recognised fact has important implications for health, society and economics. The most evident is in the number of people who report activity limitations, which grows significantly with age as clearly shown in the following chart. Activity limitations have an adverse effect on a person's productivity, on the quality of her social relations and, ultimately, on her quality of life. Policy makers confronted with a problem of challenging complexity: how to develop an effective strategy to fight the physical and cognitive decline of older adults in the face of ever shrinking financial resources for health care and social services.

In this context, technology can be of considerable help to care–givers to extend the range and the efficacy of their actions. The ACANTO project (http://www.ict-acanto.eu) aims to develop a portfolio of technical solution that can serve this purpose. More specifically, our goal is to spur older adults into a sustainable and regular level of physical exercise under the guidance and the supervision of their carers.

The key elements of ACANTO are a robotic friend (the FriWalk) that sup-ports the user in the execution of daily activities that require physical exercise and an intelligent system that recommends activites that a senior user perceives as compelling and rewarding.

The FriWalk takes the form of a standard walking assistant, but it is in fact an intelligent robot that is able to localise itself, to sense the surrounding environment, to plan a course of action that suits the user needs and to guide the user along safe routes. The FriWalk is also a personal trainer that can support the user in the execution of a training programme, monitor the motion of the user in search of muscular or gait problems and report them into the user profile (that can be inspected by doctors and physicians).

The second key idea of ACANTO is that physical exercise is actually "concealed" within compelling activities (such as shopping, taking walks in museums and exhibitions etc.). Such activities have a social dimension (they are proposed to group of users) and are chosen based on the interest and on the past experiences of the user. At the heart of the recommendation system there is a social network which is created and developed by primarily using information collected by the FriWalk using "physical" observations on her behaviour and on her emotional state. For this reason, we call this social network "cyberphysical".

This project aims at developping an autonomous system to drive groups of citizens with respect to point of interest. Those citizens are supposed to communicate, and one of the objective of Tamis is to build a robust and secure system to guarantee this communication. Axel Legay and Olivier Zendra are the permanent researchers of Tamis involved in this project. The project supports two postdocs in Tamis.

*9.3.1.2. DIVIDEND*

**Participant:** Laurent Morin.

Start: 2014. End: 2017.

The DIVIDEND project (http://www.chistera.eu/projects/dividend) attacks the data centre energy efficiency bottleneck through vertical integration, specialisation, and cross-layer optimization. Our vision is to present heterogeneous data centres, combining CPUs, GPUs, and task-specific accelerators, as a unified entity to the application developer and let the runtime optimize the utilization of the system resources during task execution. DIVIDEND embraces heterogeneity to dramatically lower the energy per task through extensive hardware specialisation while maintaining the ease of programmability of a homogeneous architecture. To lower communication latency and energy, DIVIDEND refers a lean point-to-point messaging fabric over complex connection-oriented network protocols. DIVIDEND addresses the programmability challenge by adapting and extending the industry-led heterogeneous systems architecture programming language and runtime initiative to account for energy awareness and data movement. DIVIDEND provides for a cross-layer energy optimization framework via a set of APIs for energy accounting and feedback between hardware, compilation, runtime, and application layers. The DIVIDEND project will usher in a new class of vertically integrated data centres and will take a first stab at resolving the energy crisis by improving the power usage effectiveness of data centres.

Laurent Morin from Tamis is involved in this project

*9.3.1.3. EMC$^2$*

**Participants:** Axel Legay, Olivier Zendra.

Start: 2014. End: 2017.

EMC$^2$ (Embedded Multi-Core systems for Mixed Criticality applications in dynamic and changeable real-time environments https://www.artemis-emc2.eu) is an ARTEMIS Joint Undertaking project in the Innovation Pilot Programme 'Computing platforms for embedded systems' (AIPP5). Embedded systems are the key innovation driver to improve almost all mechatronic products with cheaper and even new functionalities. They support today's information society as inter-system communication enabler. A major industrial challenge arises from the need to face cost efficient integration of different applications with different levels of safety and security on a single computing platform in an open context. EMC$^2$ finds solutions for dynamic adaptability in open systems, provides handling of mixed criticality applications under real-time conditions, scalability and utmost flexibility, full scale deployment and management of integrated tool chains, through the entire lifecycle. The objective of EMC$^2$ is to establish Multi-Core technology in all relevant Embedded Systems domains. EMC$^2$ is a project of 99 partners of embedded industry and research from 19 European countries with an effort of about 800 person years and a total budget of about 100 million Euro.

EMC2 (2014–2017) is at the border between formal methods and security. We in Tamis are mainly using the fundings to develop the Plasma toolset that is used by our statistical model checking and symbolic model checking tools. The permanent members of Tamis who are involved are Axel Legay and Olivier Zendra. The project was initiated during the lifetime of the ESTASYS.Inria team.

*9.3.1.4. ENABLE-S3*

**Participants:** Axel Legay, Jean-Louis Lanet.

Start: 2016. End: 2019.

The objective of ENABLE-S3 (http://www.enable-s3.eu) is to establish cost-efficient cross-domain virtual and semi-virtual V&V platforms and methods for ACPS. Advanced functional, safety and security test methods will be developed in order to significantly reduce the verification and validation time but preserve the validity of the tests for the requested high operation range. ENABLE-S3 aspires to substitute today's physical validation and verification efforts by virtual testing and verification, coverage-oriented test selection methods and standardization. ENABLE-S3 is use-case driven; these use cases represent relevant environments and scenarios. Each of the models, methods and tools integrated into the validation platform will be applied to at least one use case (under the guidance of the V&V methodology), where they will be validated (TRL 5) and their usability demonstrated (TRL6). Representative use cases and according applications provide the base for the requirements of methods and tools, as well as for the evaluation of automated systems and respective safety.

This project is industry driven and has the objective of designing new technologies for autonomous transportation, including to secure them. Tamis tests its results on the case studies of the project. Axel Legay and Jean-Louis Lanet are involved in this project. The project supports one postdoc in Tamis starting in 2017.

## 9.4. International Research Visitors

### *9.4.1. Visits of International Scientists*

- Clémentine MAURICE (Graz University of Technology, Institute of Applied Information Processing and Communications, Austria) visited Tamis and also gave a talk on Reverse-engineering CPUs for fun and profit.

### *9.4.2. Visits to International Teams*

- Axel Legay stayed at Namur University, Belgium.
- Axel Legay stayed at University of Limerick, Ireland.
- Axel Legay and Sean Sedwards stayed at Aalborg University, Denmark.
- Axel Legay, Fabrizio Biondi and Thomas Given-Wilson stayed at John Hopkins University, USA.

# 10. Dissemination

## 10.1. Promoting Scientific Activities

### *10.1.1. Scientific Events Organisation*

*10.1.1.1. General Chair, Scientific Chair*

- Axel Legay has been the general chair for the 11th International Conference on Risks and Security of Internet and Systems

*10.1.1.2. Member of Organizing Committees*

- Axel Legay has been organizing the ICT-Energy Science Conference 2016.

### *10.1.2. Scientific Events Selection*

*10.1.2.1. Member of Conference Steering Committees*

- Olivier Zendra is a founder and a member of the Steering Committee of ICOOOLPS (International Workshop on Implementation, Compilation, Optimization of OO Languages, Programs and Systems)

*10.1.2.2. Chair of Conference Program Committees*

- Axel Legay has been the chair for the 14'International Symposium on Automated Technology for Verification and Analysis

*10.1.2.3. Member of Conference Program Committees*

- Axel Legay has been PC member for ASE, MEMOCODE, FASE, RV, SPLC, FORMATS, FORMALIZE, SETTA,
- Jean-Louis Lanet has been PC member of Cardis 2016, 15th Smart Card Research and Advanced Application Conference, Crisis 2016 The Eleventh International Conference on Risks and Security of Internet and Systems CRiSIS 2016, GraMSec'16, The Third International Workshop on Graphical Models for Security, June 27th Lisbon, Portugal Ressi 2016, Rendez-Vous de la Recherche et de l'Enseignement de la Sécurité des Systèmes d'Information, Toulouse, France, Afadl2016, 15èmes Journées Francophones Internationales sur les Approches Formelles dans l'Assistance au Développement de Logiciels.
- Olivier Zendra has been PC member of PEC 2016 (International Conference on Pervasive and Embedded Computing)

### *10.1.3. Journal*

*10.1.3.1. Member of the Editorial Boards*

- Axel Legay is a funder and member of the editorial board of "Foundations for Mastering Changes" journal.

*10.1.3.2. Reviewer - Reviewing Activities*

- Axel Legay has been reviewer for TCS, TSE, Information and Computation.

### *10.1.4. Invited Talks*

- Axel Legay has been an invited speaker for the 10th International Workshop on Reachability Problems.
- Axel Legay has been an invited speaker for the ICT-Energy Science Conference 2016.
- The Wheel of Fault Injection, J.-L Lanet, Workshop Sertif, Grenoble, October 2016.
- Christian Grothoff. "Enabling Secure Web Payments with GNU Taler". Keynote at SPACE 2016 (December).
- Christian Grothoff. "Anonymous Payment Systems" at MAPPING Second General Assembly, Prague, 2016.
- Christian Grothoff. "Netzwerksicherheit: Probleme und Lösungsansätze" at NPO Kongress, Wien, 2016.
- Christian Grothoff. "The GNU Name System: A Public Key Infrastructure for Social Movements in the Age of Universal Surveillance" at Johns Hopkins University, Baltimore, USA, 2016.
- Christian Grothoff. "GNU Taler" at the Free Software Foundation Fellowship Meeting, Düsseldorf, 2016.
- Christian Grothoff. "The GNU Name System: A clean-slate solution to the DNS security and privacy nightmare" at Journée du Conseil scientifique de l'Afnic, Paris, 2016.
- Christian Grothoff. "GNU Taler: A privacy-preserving online payment system for libre society" at CubaConf, Havana, 2016.
- Jeffrey Burdges. "GNU Taler" at the Internet Freedom Festival, 2016.
- Jeffrey Burdges. Preliminary report "Xolotl A compact mixnet format with stronger forward secrecy and hybrid anonymity" at the GNU Hacker Meeting, 2016.
- Florian Dold presented "GNU Taler – Privacy preserving payments for the web" at the GNU Hacker Meeting, 2016.

- Jeffrey Burdges. Panel on “Privacy-preserving decentralization: what challenges are lying ahead?” at the ECRYPT 2016 Workshop on Strategic Research Challenges for Privacy Technologies.
- Christian Grothoff. Panel on "Innovation, Complexity, Risk and Trust" at MAPPING Second General Assembly, Prague, 2016.

### *10.1.5. Scientific Expertise*

- Axel Legay is an expert for the Wallonie Government.
- Axel Legay is a member of Inria’s evaluation committee. He participated to the CR2 and CR1 juries for Lille Center.
- Axel Legay has been in the jury for the chair on cyber security at CentralSupelec.
- Jeffrey Burdges, Christian Grothoff, and Florian Dold have been involved in the W3C Payments Working Group, primarily contributing security and privacy comments on their evolving standard.
- Olivier Zendra is a CIR expert for the MENESR.
- Olivier Zendra is a member of Inria’s evaluation committee. He participated to the CR2 jury for Grenoble Center, to the national CR1 promotion jury, and to the workgroup on the creation of the PACAP team of Inria Rennes.
- Olivier Zendra is a member of the editorial board and co-author of the “HiPEAC Vision” [69]

### *10.1.6. Research Administration*

- Axel Legay is a member of Inria’s evaluation committee.
- Axel Legay is the Representative for non-permanent staff committees (in charge of postdocs).
- Olivier Zendra is a member of Inria’s evaluation committee.
- Olivier Zendra was a member of Inria’s Parity and Equal Opportunities committee.
- Olivier Zendra is a member of Inria’s worksgroup on Inria’s social barometer.
- Olivier Zendra was a member of Inria’s CNHSCT.
- Olivier Zendra was Head of Inria Nancy’s IES Committee (formerly IST).

## 10.2. Teaching - Supervision - Juries

### *10.2.1. Teaching*

- Master : Axel Legay, Introduction au Model Checking, 36, M2, Université de Bretagne Sud, France
- Master : Axel Legay, Introduction à l’analyse de risques, M2, Université de Bretagne Sud, France

### *10.2.2. Supervision*

- PhD : Aymerick Savary, De la génération de suites de test à partir de modèles formels, University of Sherbrook and University of Limoges, 30th June 2016, Marc Frappier, Jean-Louis Lanet
- PhD : Tiana Razafindralambo, Attaques combinées sur appareil mobiles, University of Limoges, November 2016, Christophe Clavier, Jean-Louis Lanet
- PhD : Neal Walfield, Location prediction for context-aware applications, Johns Hopkins, 4th October 2016, Christian Grothoff
- PhD in progress : Kevin Bukasa, Démarrage sécurisé, 2015, Jean-Louis Lanet and Axel Legay
- PhD in progress : Mounir Chadli (Rennes 1), On Scheduling and SMC, December 2014, Axel Legay and Saddek Bensalem.
- PhD in progress : Olivier Descourbe, On Code Obfuscation, October 2016, Axel Legay and Fabrizio Biondi.
- PhD in progress : Mike Enescu, On Symbolic Execution for Malware Detection, October 2016, Axel Legay and Flavio Oquendo and Fabrizio Biondi.

- PhD in progress : Alexandre Gonsalvez, On Obfuscation via crypto primitives, April 2016, Axel Legay and Caroline Fontaine.
- PhD in progress : Nisrine Jafri (Rennes1), On fault Injection detection with MC of Binary code, December 2015, Axel Legay and Jean-Louis Lanet.
- PhD in progress : Razika Lounas, Validation des spécifications formelles de la mise à jour dynamique des applications Java Card, 2010, Mohamed Mezghiche and Jean-Louis Lanet
- PhD in progress : Aurélien Palisse, Observabilité de codes hostiles, 2015, Jean-Louis Lanet
- PhD in progress : Aurélien Trulla, Caractérisation de malware Android par suivi de flux d'information et nouvelles techniques d'évasion, 2016, Valerie Viet Triem Tong and Jean-Louis Lanet
- PhD in progress : Tristan Ninet (Rennes 1), Vérification formelle d'une implémentation de la pile protocolaire IKEv2, December 2016, Axel Legay, Romaric Maillard and Olivier Zendra

### *10.2.3. Juries*

- Axel Legay has been a referee for the PhD defense of Najah Ben Said (University of Grenoble Alpes).
- Axel Legay has been a member of the jury for the PhD defense of Zaruhi Aslanyan (DTU Denmark).
- Jean-Louis Lanet has been a referee for the PhD defense of Pierre Belgarric (Télécom ParisTech).
- Jean-Louis Lanet has been a referee for the PhD defense of Louis Dureuil (University of Grenoble Alpes).
- Jean-Louis Lanet has been a referee for the PhD defense of Gabriel Risterucci (University of Aix Marseille).
- Jean-Louis Lanet has been a member of the jury for the PhD defense of Benoît Morgan (University of Toulouse).
- Jean-Louis Lanet has been a member of the jury for the PhD defense of Najah Ben Said (University of Grenoble Alpes).
- Olivier Zendra has been a co-referee for the PhD defense of Rabah Laouadi (University of Montpellier).

## 10.3. Popularization

- Vulnerability Prediction Against Fault Attacks , N. Jafri, A. Legay, J.-L. Lanet, Ercim news 106, 2016
- Skyfall : Tombé du ciel, J.-L. Lanet, Interstices, 2016 In this publication we revisit the movie Skyfall and explain on which scientific background rely some elements of the movie.
- FIC 2016 Internet des objets : la nouvelle fragilité ? We have been invited to participate at a panel with layers, IoT designer to discuss the security of the IoT.
- Atlantico, Et si les objets connectés étaient la plus grande faille qu'entreprises et particuliers pouvaient offrir aux hackers ? January 2016. In this interview we explain that the security is not the main concern of low end IoT, which is not the case with high end IoT.

# 11. Bibliography

## Publications of the year

### Articles in International Peer-Reviewed Journals

[1] F. BIONDI, T. GIVEN-WILSON, A. LEGAY. *Attainable Unconditional Security for Shared-Key Cryptosystems*, in "Information Sciences", September 2016, https://hal.inria.fr/hal-01378640

[2] N. JANATI, G. BOUFFARD, J.-L. LANET, S. ELHAJJI. *Trust Can be Misplaced*, in "Journal of Cryptographic Engineering", 2016 [*DOI :* 10.1007/s13389-016-0142-5], https://hal.inria.fr/hal-01405463

[3] C. JEGOUREL, A. LEGAY, S. SEDWARDS. *Command-based importance sampling for statistical model checking*, in "Theoretical Computer Science", 2016, vol. 649, pp. 1 - 24, https://hal.inria.fr/hal-01387299

[4] J. H. KIM, A. LEGAY, L.-M. TRAONOUEZ, A. BOUDJADAR, U. NYMAN, K. G. LARSEN, I. LEE, J.-Y. CHOI. *Optimizing the Resource Requirements of Hierarchical Scheduling Systems*, in "ACM SIGBED Review", June 2016, vol. 13, n° 3, pp. 41 - 48 [*DOI :* 10.1145/2983185.2983192], https://hal.inria.fr/hal-01406459

[5] T. T. H. LE, R. PASSERONE, U. FAHRENBERG, A. LEGAY. *A Tag Contract Framework for Modeling Heterogeneous Systems*, in "Science of Computer Programming", January 2016, https://hal.inria.fr/hal-01406446

[6] T. T. H. LE, R. PASSERONE, U. FAHRENBERG, A. LEGAY. *Contract-Based Requirement Modularization via Synthesis of Correct Decompositions*, in "Theory of Computing Systems", May 2016, vol. 15 [*DOI :* 10.1145/2885752], https://hal.inria.fr/hal-01406481

[7] A. LEGAY, L.-M. TRAONOUEZ. *Statistical Model Checking with Change Detection*, in "Foundations for Mastering Change", September 2016, https://hal.archives-ouvertes.fr/hal-01242138

[8] F. MOZHDEH, J.-L. LANET. *Chronicle of a Java Card Death*, in "Journal of Computer Virology and Hacking Techniques", 2016 [*DOI :* 10.1007/s11416-016-0276-0], https://hal.inria.fr/hal-01385197

[9] A. NOURI, M. BOZGA, A. MOLNOS, A. LEGAY, S. BENSALEM. *ASTROLABE: A Rigorous Approach for System-Level Performance Modeling and Analysis*, in "ACM Transactions on Embedded Computing Systems (TECS)", May 2016 [*DOI :* 10.1145/2885498], https://hal.inria.fr/hal-01406474

### Invited Conferences

[10] A. LEGAY, S. SEDWARDS, L.-M. TRAONOUEZ. *Rare Events for Statistical Model Checking: An Overview*, in "Reachability Problems", Aalborg, Denmark, September 2016, vol. 9899, pp. 23 - 35, https://hal.inria.fr/hal-01387406

### International Conferences with Proceedings

[11] A. ARNOLD, M. BALEANI, A. FERRARI, M. MARAZZA, V. SENNI, A. LEGAY, J. QUILBEUF, C. ETZIEN. *An Application of SMC to continuous validation of heterogeneous systems*, in "Simutools 2016 - Ninth EAI International Conference on Simulation Tools and Techniques", Prague, Czech Republic, August 2016, https://hal.inria.fr/hal-01390487

[12] F. BIONDI, A. LEGAY. *Security and Privacy of Protocols and Software with Formal Methods*, in "7th International Symposium on Leveraging Applications of Formal Methods, Verification and Validation", Kommeno, Greece, October 2016, https://hal.inria.fr/hal-01378645

[13] J. BURDGES, F. DOLD, C. GROTHOFF, M. STANISCI. *Taler: Usable, privacy-preserving payments for the Web*, in "HotPETS 2016 - Workshop on Hot Topics in Privacy Enhancing Technologies", Darmstadt , Germany, June 2016, https://hal.inria.fr/hal-01398201

[14] E. CAVALCANTE, J. QUILBEUF, L.-M. TRAONOUEZ, F. OQUENDO, T. BATISTA, A. LEGAY. *Statistical Model Checking of Dynamic Software Architectures*, in "ECSA 2016 - 10th European Conference on Software Architecture", Copenhague, Denmark, Proceedings of the 2016 European Conference of Software Architecture, November 2016, https://hal.inria.fr/hal-01390707

[15] M. CHADLI, J. H. KIM, A. LEGAY, L.-M. TRAONOUEZ, S. NAUJOKAT, B. STEFFEN, K. G. LARSEN. *A Model-Based Framework for the Specification and Analysis of Hierarchical Scheduling Systems*, in "FMICS-AVoCS", Pise, Italy, Critical Systems: Formal Methods and Automated Verification, Springer, September 2016, vol. 9933, pp. 133 - 141 [*DOI :* 10.1007/978-3-319-45943-1_9], https://hal.archives-ouvertes.fr/hal-01241681

[16] D. COUROUSSÉ, T. BARRY, B. ROBISSON, P. JAILLON, O. POTIN, J.-L. LANET. *Runtime Code Polymorphism as a Protection Against Side Channel Attacks*, in "10th IFIP WG 11.2 International Conference, WISTP 2016", Heraklion, Greece, Information Security Theory and Practice, September 2016, vol. 9895, pp. 136-152 [*DOI :* 10.1007/978-3-319-45931-8_9], https://hal-emse.ccsd.cnrs.fr/emse-01372223

[17] P. R. D'ARGENIO, A. HARTMANNS, A. LEGAY, S. SEDWARDS. *Statistical Approximation of Optimal Schedulers for Probabilistic Timed Automata*, in "Integrated Formal Methods", Reykjavik, Iceland, June 2016, vol. 9681, pp. 99-114, https://hal.inria.fr/hal-01387362

[18] X. DEVROEY, G. PERROUIN, A. LEGAY, P.-Y. SCHOBBENS, P. HEYMANS. *Search-based Similarity-driven Behavioural SPL Testing*, in "VaMoS '16 - Tenth International Workshop on Variability Modelling of Software-intensive Systems", Salvador, Brazil, ACM, January 2016, pp. 89 - 96 [*DOI :* 10.1145/2866614.2866627], https://hal.inria.fr/hal-01406585

[19] X. DEVROEY, G. PERROUIN, M. PAPADAKIS, A. LEGAY, P.-Y. SCHOBBENS, P. HEYMANS. *Featured model-based mutation analysis*, in "ICSE '16 - 38th International Conference on Software Engineering", Austin, United States, May 2016, pp. 655 - 666 [*DOI :* 10.1145/2884781.2884821], https://hal.inria.fr/hal-01406512

[20] O. GADYATSKAYA, R. RYDHOF HANSEN, K. G. LARSEN, A. LEGAY, M. C. OLESEN, D. B. POULSEN. *Modelling Attack-defense Trees Using Timed Automata*, in "FORMATS 2016 -14th International Conference on Formal Modelling and Analysis of Timed Systems", Quebec City, Canada, August 2016, pp. 35 - 50 [*DOI :* 10.1007/978-3-319-44878-7_3], https://hal.inria.fr/hal-01406706

[21] Á. GARCÍA-RECUERO, J. BURDGES, C. GROTHOFF. *Privacy-Preserving Abuse Detection in Future Decentralised Online Social Networks*, in "11th International ESORICS Workshop in Data Privacy Management, DPM 2016", Heraklion, Crete, Greece, G. LIVRAGA, V. TORRA, A. ALDINI, F. MARTINELLI, N. SURI (editors), Springer Lecture Notes in Computer Science (LNCS) series, Springer, September 2016, vol. 9963, pp. 78-93 [*DOI :* 10.1007/978-3-319-47072-6_6], https://hal.inria.fr/hal-01355951

[22] T. GIVEN-WILSON, A. LEGAY. *On the Expressiveness of Symmetric Communication*, in "Theoretical Aspects of Computing – ICTAC 2016", Taipei, Taiwan, October 2016, vol. 9965, pp. 139-157 [*DOI :* 10.1007/978-3-319-46750-4_9], https://hal.inria.fr/hal-01241839

[23] A. HEUSER, S. PICEK, S. GUILLEY, N. MENTENS. *Side-channel Analysis of Lightweight Ciphers: Does Lightweight Equal Easy?*, in "RFIDSec 2016: 12th Workshop on RFID and IoT Security", Hong Kong, Hong Kong SAR China, November 2016, https://hal.inria.fr/hal-01402238

[24] Y. KAWAMOTO, F. BIONDI, A. LEGAY. *Hybrid Statistical Estimation of Mutual Information for Quantifying Information Flow*, in "FM 2016 - 21st International Symposium on Formal Methods", Limassol, Cyprus, November 2016, https://hal.inria.fr/hal-01378675

[25] J. H. KIM, A. LEGAY, L.-M. TRAONOUEZ, M. ACHER, S. KANG. *A Formal Modeling and Analysis Framework for Software Product Line of Preemptive Real-Time Systems*, in "Symposium on Applied Computing", Pise, Italy, Proceedings of the 31st Annual ACM Symposium on Applied Computing, ACM, April 2016, pp. 1562 - 1565 [*DOI :* 10.1145/2851613.2851977], https://hal.archives-ouvertes.fr/hal-01241673

[26] K. G. LARSEN, A. LEGAY. *On the Power of Statistical Model Checking*, in "7th International Symposium, ISoLA 2016", Corfu, Greece, October 2016, pp. 843 - 862 [*DOI :* 10.1007/978-3-319-47169-3_62], https://hal.inria.fr/hal-01406537

[27] H. LE BOUDER, T. BARRY, D. COUROUSSÉ, J.-L. LANET, R. LASHERMES. *A Template Attack Against VERIFY PIN Algorithms*, in "SECRYPT 2016", Lsbonne, Portugal, July 2016, pp. 231 - 238 [*DOI :* 10.5220/0005955102310238], https://hal.inria.fr/hal-01383143

[28] H. LE BOUDER, R. LASHERMES, Y. LINGE, G. THOMAS, J.-Y. ZIE. *A Multi-Round Side Channel Attack on AES using Belief Propagation*, in "FPS 2016", Québec, Canada, October 2016, https://hal.inria.fr/hal-01405793

[29] A. LEGAY, S. SEDWARDS, L.-M. TRAONOUEZ. *Plasma Lab: A Modular Statistical Model Checking Platform*, in "ISoLA", Corfou, Greece, Leveraging Applications of Formal Methods, Verification and Validation: Foundational Techniques, Springer, October 2016, vol. 9952, pp. 77 - 93 [*DOI :* 10.1007/978-3-319-47166-2_6], https://hal.inria.fr/hal-01387435

[30] R. LOUNAS, N. JAFRI, A. LEGAY, M. MEZGHICHE, J.-L. LANET. *A Formal Verification of Safe Update Point Detection in Dynamic Software Updating*, in "International Conference on Risks and Security of Internet and Systems", Roscoff, France, LNCS, Telecom Bretagne, September 2016, https://hal.inria.fr/hal-01405467

[31] R. LOUNAS, N. JAFRI, A. LEGAY, M. MEZGHICHE, J.-L. LANET. *A Formal Verification of Safe Update Point Detection in Dynamic Software Updating*, in "The 11th International Conference on Risks and Security of Internet and Systems (CRiSIS 2016)", Roscoff, France, September 2016, https://hal.inria.fr/hal-01400051

[32] A. MESBAH, L. REGNAUD, J.-L. LANET, M. MEZGHICHE. *The Hell Forgery, Polymorphic Codes Shoot Again*, in "15th Smart Card Research and Advanced Application Conference", Cannes, France, Aurélien Francillon, November 2016, https://hal.inria.fr/hal-01385202

[33] F. MOZHDEH, J.-L. LANET. *Paper Tiger, an endless fight*, in "SecITC 9th International Conference on Security for Information Technology and Communications", Bucarest, Romania, Innovative Security Solutions for Information Technology and Communications, June 2016, vol. LNCS, n$^{o}$ 10006 [*DOI :* 10.1007/978-3-319-47238-6], https://hal.inria.fr/hal-01385201

[34] A. MRABET, N. EL-MRABET, R. LASHERMES, J.-B. RIGAUD, B. BOUALLEGUE, S. MESNAGER, M. MACHHOUT. *High-performance Elliptic Curve Cryptography by Using the CIOS Method for Modular Multiplication*, in "CRiSIS 2016", Roscoff, France, September 2016, https://hal.inria.fr/hal-01383162

[35] V. C. NGO, A. LEGAY, V. JOLOBOFF. *PSCV: A Runtime Verification Tool for Probabilistic SystemC Models*, in "CAV 2016 - 28th International Conference on Computer Aided Verification", Toronto, Canada, S.

CHAUDHURI, A. FARZAN (editors), LNCS - Lecture Notes in Computer Science, Springer, July 2016, vol. 9779, pp. 84 - 91 [*DOI :* 10.1007/978-3-319-41528-4_5], https://hal.inria.fr/hal-01406488

[36] A. NOURI, M. BOZGA, A. LEGAY, S. BENSALEM. *Performance Evaluation of Complex Systems Using the SBIP Framework*, in "he International Conference on Verification and Evaluation of Computer and Communication Systems (VECoS)", Tunis, Tunisia, October 2016, https://hal.inria.fr/hal-01406591

[37] R. OLAECHEA, U. FAHRENBERG, J. M. ATLEE, A. LEGAY. *Long-term average cost in featured transition systems*, in "SPLC '16 - 20th International Systems and Software Product Line Conference", Beijing, China, ACM, September 2016, pp. 109 - 118 [*DOI :* 10.1145/2934466.2934473], https://hal.inria.fr/hal-01406541

[38] R. OLAECHEA, U. FAHRENBERG, M. JOANNE, A. LEGAY. *Long-Term Average Cost in Featured Transition Systems*, in "the 20th International Systems and Software Product Line Conference", Beijing, China, September 2016, https://hal.inria.fr/hal-01406606

[39] F. OQUENDO, J. BUISSON, E. LEROUX, G. MOGUÉROU, J. QUILBEUF. *The SoS Architect Studio: Toolchain for the Formal Architecture Description and Analysis of Software-intensive Systems-of-Systems with SosADL*, in "Proceedings of the ECSA International Colloquium on Software-intensive Systems-of-Systems (SiSoS)", Copenhagen, Denmark, November 2016, https://hal.archives-ouvertes.fr/hal-01443130

[40] G. PERROUIN, M. AMRANI, M. ACHER, B. COMBEMALE, A. LEGAY, P.-Y. SCHOBBENS. *Featured model types: Towards Systematic Reuse in Modelling Language Engineering*, in "MiSE '16 - 8th International Workshop on Modeling in Software Engineering", New york, United States, ACM, May 2016, pp. 1 - 7 [*DOI :* 10.1145/2896982.2896987], https://hal.inria.fr/hal-01406507

[41] J. QUILBEUF, E. CAVALCANTE, L.-M. TRAONOUEZ, F. OQUENDO, T. BATISTA, A. LEGAY. *A Logic for the Statistical Model Checking of Dynamic Software Architectures*, in "ISoLA", Corfou, Greece, Leveraging Applications of Formal Methods, Verification and Validation: Foundational Techniques, Springer, October 2016, vol. 9952, pp. 806 - 820 [*DOI :* 10.1007/978-3-319-47166-2_56], https://hal.inria.fr/hal-01387429

[42] M. H. TER BEEK, A. LEGAY, A. LLUCH LAFUENTE, A. VANDIN. *Statistical Model Checking for Product Lines* , in "7th International Symposium, ISoLA 2016", Corfu, Greece, October 2016, https://hal.inria.fr/hal-01406531

[43] N. H. WALFIELD, J. L. GRIFFIN, C. GROTHOFF. *A Quantitative Analysis of Cell Tower Trace Data for Understanding Human Mobility and Mobile Networks*, in "6th International Workshop on Mobile Entity Localization, Tracking and Analysis (MELT)", San Francisco, United States, October 2016 [*DOI :* 10.1145/1235], https://hal.inria.fr/hal-01378622

### Conferences without Proceedings

[44] M. BOIZARD, J.-L. LANET. *Internet des objets : la nouvelle fragilité ? : Atelier*, in "Forum international de la cybersécurité", Lille, France, Région Nord-Pas de Calais, Euratechnologies, la Gendarmerie nationale, CEIS, January 2016, https://halshs.archives-ouvertes.fr/halshs-01259237

[45] D. COUROUSSÉ, J.-L. LANET, B. ROBISSON, T. BARRY, P. JAILLON. *COGITO: Génération de code au runtime pour la sécurité des systèmes embarqué*, in "Ressi 2016", Toulouse, France, May 2016, https://hal.inria.fr/hal-01405776

[46] J. DUCHÊNE, C. LE GUERNIC, E. ALATA, V. NICOMETTE, M. KAÂNICHE. *Protocol reverse engineering: Challenges and obfuscation*, in "The 11th International Conference on Risks and Security of Internet and Systems", Roscoff, France, September 2016, https://hal.inria.fr/hal-01388060

[47] M. GRAA, F. CUPPENS, N. CUPPENS-BOULAHIA, J.-L. LANET. *Tracking explicit and control flows in Java and native Android apps code,*, in "ICISSP 2016 : 2nd International Conference on Information Systems Security and Privacy,", Roma, Italy, February 2016, https://hal.inria.fr/hal-01385196

[48] Á. GARCÍA-RECUERO. *Discouraging Abusive Behavior in Privacy-Preserving Online Social Networking Applications*, in "In 25th International World Wide Web Conference", Montréal, Canada, April 2016, vol. WWW Companion Volume, https://hal.inria.fr/hal-01272654

[49] C. JEGOUREL, K. G. LARSEN, A. LEGAY, M. MIKUČIONIS, D. B. POULSEN, S. SEDWARDS. *Importance Sampling for Stochastic Timed Automata*, in "Dependable Software Engineering: Theories, Tools, and Applications", Beijing, China, November 2016, https://hal.inria.fr/hal-01387293

[50] V. C. NGO, A. LEGAY, J. QUILBEUF. *Statistical Model Checking for SystemC Models*, in "High Assurance Systems Engineering Symposium", Orlando, Florida, United States, January 2016, https://hal.inria.fr/hal-01238162

[51] A. PALISSE, A. DURAND, J.-L. LANET. *Malware'O'Matic a platform to analyze Malware*, in "French Japanese workshop on CyberSecurity", Rennes, France, Inria, September 2016, https://hal.inria.fr/hal-01405636

[52] A. PALISSE, J.-L. LANET. *Analyse et détection de logiciels malveillants,* , in "Ressi 2016", Toulouse, France, May 2016, https://hal.inria.fr/hal-01405627

[53] A. PALISSE, H. LE BOUDER, C. LE GUERNIC, A. LEGAY, J.-L. LANET. *Ransomware and the Legacy Crypto API*, in "The 11th International Conference on Risks and Security of Internet and Systems", Roscoff, France, September 2016, https://hal.inria.fr/hal-01388056

### Scientific Books (or Scientific Book chapters)

[54] S. CHAUMETTE, J. H. KIM, K. NAMUDURI, J. P. STERBENZ (editors). *UAV Networks and Communications*, Cambridge University Press, 2016, https://hal.archives-ouvertes.fr/hal-01391820

### Books or Proceedings Editing

[55] F. OQUENDO, M. ALI BABAR, K. DRIRA, A. LEGAY (editors). *Proceedings of the European Colloquium on Software-intensive Systems-of-Systems (ECSoS 2016)*, ACM DL, Copenhagen, Denmark, December 2016, https://hal.archives-ouvertes.fr/hal-01445339

[56] F. OQUENDO, K. DRIRA, A. LEGAY, T. BATISTA (editors). *Proceedings of the 1st ACM SAC Conference Track on Software-intensive Systems-of-Systems (SiSoS 2017): 32nd ACM SIGAPP Symposium On Applied Computing*, ACM, Marrakesh, Morocco, April 2017, https://hal.archives-ouvertes.fr/hal-01445350

### Scientific Popularization

[57] F. BIONDI, S. JOSSE, A. LEGAY. *Bypassing Malware Obfuscation with Dynamic Synthesis*, in "ERCIM News", September 2016, n$^{o}$ 106, https://hal.inria.fr/hal-01378662

[58] C. GROTHOFF, J. PORUP. *The NSA's SKYNET program may be killing thousands of innocent people*, in "Ars Technica", February 2016, https://hal.inria.fr/hal-01278193

[59] N. JAFRI, A. LEGAY, J.-L. LANET. *Vulnerability Prediction Against Fault Attacks*, in "ERCIM News", July 2016, https://hal.inria.fr/hal-01394973

[60] J.-L. LANET. *Skyfall : Tombé du ciel*, in "Interstices", June 2016, https://hal.inria.fr/hal-01352807

[61] H. LE BOUDER. *Des attaques informatiques utilisant la physique*, in "Interstices", November 2016, https://hal.inria.fr/hal-01408630

### Other Publications

[62] F. BIONDI, M. CHADLI, T. GIVEN-WILSON, A. LEGAY. *Information Leakage as a Scheduling Resource*, October 2016, working paper or preprint, https://hal.inria.fr/hal-01382052

[63] F. BIONDI, T. GIVEN-WILSON, A. LEGAY. *Attainable Unconditional Security for Shared-Key Cryptosystems*, April 2016, working paper or preprint, https://hal.inria.fr/hal-01233185

[64] Q. CAPPART, C. LIMBRÉE, P. SCHAUS, J. QUILBEUF, L.-M. TRAONOUEZ, A. LEGAY. *Verification of interlocking systems using statistical model checking*, November 2016, working paper or preprint, https://hal.inria.fr/hal-01398649

[65] U. FAHRENBERG, A. LEGAY. *A Linear-Time–Branching-Time Spectrum of Behavioral Specification Theories*, October 2016, We propose behavioral specification theories for most equivalences in the linear-time–branching-time spectrum, https://hal.inria.fr/hal-01406603

[66] T. GIVEN-WILSON, N. JAFRI, J.-L. LANET, A. LEGAY. *An Automated Formal Process for Detecting Fault Injection Vulnerabilities in Binaries*, January 2017, working paper or preprint, https://hal.inria.fr/hal-01400283

[67] A. HEUSER, S. PICEK, S. GUILLEY, N. MENTENS. *Side-channel Analysis of Lightweight Ciphers: Current Status and Future Directions*, October 2016, Lightweight Cryptography Workshop 2016, https://hal.inria.fr/hal-01407264

[68] Y. KAWAMOTO, F. BIONDI, A. LEGAY. *Hybrid Statistical Estimation of Mutual Information for Quantifying Information Flow*, September 2016, working paper or preprint, https://hal.inria.fr/hal-01241360

## References in notes

[69] M. DURANTON, K. DE BOSSCHERE, C. GAMRAT, J. MAEBE, H. MUNK, O. ZENDRA. *HiPEAC (High Performance and Embedded Architecture and Compilation) Vision 2017*, December 2016, pp. 1–137, working document, to be published early 2017, https://www.hipeac.net/v17

[70] A. LEGAY, L.-M. TRAONOUEZ. *Statistical Model Checking of Simulink Models with Plasma Lab*, in " Fourth International Workshop on Formal Techniques for Safety-Critical Systems", Paris, France, November 2015, https://hal.archives-ouvertes.fr/hal-01241249

[71] A. SAVARY, M. FRAPPIER, M. LEUSCHEL, J. LANET. *Model-Based Robustness Testing in Event-B Using Mutation*, in "Software Engineering and Formal Methods - 13th International Conference, SEFM 2015, York,

UK, September 7-11, 2015. Proceedings", R. CALINESCU, B. RUMPE (editors), Lecture Notes in Computer Science, Springer, 2015, vol. 9276, pp. 132–147, http://dx.doi.org/10.1007/978-3-319-22969-0_10